MINIMIZATION OF COLLATERAL DAMAGE IN AIRDROPS AND AIRSTRIKES

DISSERTATION

Steven P. Dillenburger, Major, USAF

AFIT/DS/ENS/12-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/DS/ENS/12-01

Minimization of Collateral Damage in Airdrops and Airstrikes

DISSERTATION

Presented to the Faculty
Department of Operational Sciences
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

Steven P. Dillenburger, B.S., B.B.A., M.S.
Major, USAF

September 2012

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/DS/ENS/12-01

Minimization of Collateral Damage in Airdrops and Airstrikes

Steven P. Dillenburger, B.S., B.B.A., M.S.
Major, USAF

Approved:

Dr. Jeffery K. Cochran
Dissertation Advisor

5 Aug 12
Date

Dr. Stephen P. Chambal
Committee Member

05AuG12
Date

Dr. James D. Cordeiro
Committee Member

6 Aug 2012
Date

Dr. Joseph J. Pignatiello
Committee Member

6 Aug 2012
Date

Accepted:

MU Thomas

M. U. Thomas, PhD
Dean, Graduate School of
Engineering and Management

5 Sep 2012
Date

For my wife

AFIT/DS/ENS/12-01

Abstract

Collateral damage presents a significant risk during air drops and airstrikes, risking citizens' lives and property, straining the relationship between the United States Air Force and host nations. This dissertation presents a methodology to determine the optimal location for making supply airdrops in order to minimize collateral damage while maintaining a high likelihood of successful recovery. A series of non-linear optimization algorithms are presented along with their relative success in finding the optimal location in the airdrop problem. Additionally, we present a quick algorithm for accurately creating the Pareto frontier in the multi-objective airstrike problem. We demonstrate the effect of differing guidelines, damage functions, and weapon employment selection which significantly alter the location of the optimal aimpoint in this targeting problem. Finally, we have provided a framework for making policy decisions in fast-moving troops-in-contact situations where observers are unsure of the nature of possible enemy forces in both finite and infinite time horizon problems. Through a recursive technique of solving this Markov decision process we have demonstrated the effect of improved intelligence and differing weights in the face of uncertain situations.

Acknowledgements

First and foremost, I would like to thank Dr. Jeffery Cochran, without whose support and dedication this dissertation would not be possible.

Thank you also to the committee members, Dr. Stephen Chambal, Dr. Joseph Pignatiello, and Dr. James Cordeiro for their instruction, time, and inputs. Thank you to the late Dr. Jim Moore, his legacy of selfless dedication to our country lives on in the minds he taught and the lives he touched. Thank you to my sponsors at the Air Force Research Laboratories, the Air Force Targeting Center, the Air Mobility Command, and the Air Armament Center for giving real-world perspectives to the work presented.

Thank you to the AFIT community for providing an incredible environment of learning; the professors, staff, and students have built a world-class research university which has enabled me to succeed. Thank you to my fellow classmates, Dr. Jason Williams, Dr. Ben Hartlage, and Dr. Jeremy Jordan, your advice and friendship these last three years has been invaluable.

Finally, I am forever indebted to my parents, Paul and Sandy. They are the role models I strive to emulate every day of my life.

Steven P. Dillenburger

Table of Contents

Page

Page

List of Figures

Figure Page

Figure Page

Minimization of Collateral Damage in Airdrops and Airstrikes

I. Introduction

1.1 Motivation

Even as advances in weapons and intelligence gathering improve U.S. military capabilites, civilian casualties and collateral damage continue to hurt the U.S. mission in the Middle East. According to sources [98] [45], over 6,000 Afghan civilians deaths can be attributed directly to U.S. and NATO military actions since the inception of the Afghanistan campaign in 2001.

Specific to the USAF, in 2006, 116 Afghan civilians were killed as a result of 13 separate OEF and ISAF bombing missions. In 2007, those numbers grew to 321 civilians in 22 bombings [38]. Aerial bombardment, which has long been the centerpiece of the U.S. strategic plan in Afghanistan, has had a devastating impact on Afghan civilians [55]. Some [90] argue that civilian casualties caused by American troops and American bombs have made the case for the insurgency.

The issue of civilian casualties has become a focal point of strategic planning for both NATO and the insurgency forces in Afghanistan. Civilian casualties often are the result of insurgents hiding among civilians or using the civilians as human shields, since they know American forces are hesitant to strike buildings in which they believe civilians are located [63]; however, that does not stop insurgents from using these incidents as rallying cries to coerce the Afghan populace. "The Taliban and Al Qaeda grasp the value of presenting themselves as defenders of the Afghan people. They distribute pamphlets in which they revile American and NATO soldiers as infidel, terrorist forces of occupation. When those same forces send planes to bomb mosques and religious schools, killing Afghan children, the Taliban do not hesitate to seize on the tragedy as proof of the validity of their propaganda - even if merciless Al Qaeda interlopers prevented those children from escaping the bombs [96]."

In November 2008, General David McKiernan, commander of U. S. Forces Afghanistan "ordered a tightening of procedures for launching airstrikes" while stating that "minimizing civilian casualties is crucial [103]." In June 2009, as his relationship with Defense Secretary Gates became strained due to the continued civilian casualties [97], he was asked to resign. His replacement, General Stanley McChrystal, in one of his first interviews upon taking the role stated, "A willingness to operate in ways minimizing casualties or damage is critical. The measure of success will not be enemy killed. It will be shielding the Afghan population from violence [1]." U.S. commanders have even gone so far as requiring troops to withdraw when possible rather than get into a protracted firefight that result in civilian casualties [39].

The issue of civilian casualties is not new to the U.S. military during the Global War on Terrorism (GWOT), but with increasing numbers of news outlets, social media forums, and personal communication devices, any mistake can be immediately consumed by people around the world, even before the facts of the scenario are fully known. Studies on collateral damage estimation from nuclear weapons were performed after World War II [88]. Kiernan and Owen [55] discuss the similarities between GWOT and Cambodian civilian casualties during the Vietnam War. Keaney [54] speaks of the intelligence issues concerning collateral damage from the Gulf War. Infamously, during the NATO campaign in Kosovo in 2000, the U.S. military mistakenly bombed the Chinese embassy in Belgrade when believing the building to be a headquarters for the Yugoslav Army [75]. Similar situations in Yugoslavia [7] [36] had resulted in buildings which have little-to-no military value being destroyed.

Even with the most modern military technology and decades of war-time experience, civilian casualties continue to plague U.S. forces and undermine the mission they seek to accomplish in the Global War on Terrorism.

1.2 Collateral Damage

The three papers presented in this dissertation explore the three categories of collateral damage and techniques or tools within each category to lower the amount of collateral risk while still achieving mission success. Chapter 2 develops a tool for minimizing collateral risk from supply airdrops based on airdrop dynamics. Chapter 3 provides a framework for understanding the trade-off between lethality on a military target and the risk to collateral objects for pre-planned airstrike missions. Chapter 4 develops guidelines for lowering civilian casualties in fast-moving troops-in-contact scenarios where limited intelligence often yields poor decision-making.

1.2.1 Airdrop Collateral Damage.

When typically thought of and reported on, collateral damage applies to weapons fired near civilians and civilian buildings. However, collateral damage also results from supply airdrops near populated areas. These airdrops often need to occur near populated areas due to safety and logistic concerns. The bundles, often weighing thousands of pounds and traveling at speeds up to 20 miles per hour, become dangerous projectiles when falling near the civilian populations. Buildings can be damaged and, in extreme cases, people have been killed [47].

Improvements in technology for airdrop platforms have vastly improved the potential accuracy of airdrop missions [5] [69]. However, the majority of airdrops are still performed by "dumb" techniques, where the bundles are not guided to the ground, but rather fall freely once exiting the back of the aircraft. These airdrop missions can often yield unrecoverable bundles when they fall in places where recovery is either impossible, such as in a lake or on a mountainside, or where recovery is dangerous, such as when bundles land miles from an operating base in hostile territory.

The official Airdrop Collateral Damage Estimation Methodology [100] notes the art and science of airdrops, when put together with sound judgment and operational considerations yield a successful drop. The guidance notes that probabilities, empirical data, and historical observations all should be considered in the planning stages of an airdrop [99]. The guidance instructs planners to ask themselves five questions regarding collateral damage during development and execution of an airdrop plan:

- Are there collateral objects within the collateral hazard area of the intended airdrop target?
- Can the functionality of the collateral objects be characterized?
- Can collateral concerns be mitigated by utilizing different parachutes/delivery methods while still achieving the desired effect?
- Are there civilians at risk by the airdrop?
- Is the collateral risk of the airdrop excessive in comparison to the expected advantage gained by the airdrop?

The collateral damage methodology presented in [100] develops an understanding of airdrop dispersion, incidental consequences (collateral risk), and mitigation techniques, but it is quick to point out that the methodology is not an exact science. The Collateral Damage Weighted Risk Assessment Tool (CDWRAT) presented in [100] uses a simple formula based on the size and location of collateral objects and the circular error probable (CE90). The equation shown below yields an overall percentage of collateral risk within the CE90.

$$A_W = (\frac{R_{long} - R_{co}}{R_{long}} + 1) \times A_{co} \tag{1}$$

where

R_{long} - CE90 semi-major radius

R_{co} - collateral object radius

A_W - weighted collateral object area

A_{co} - original collateral object area

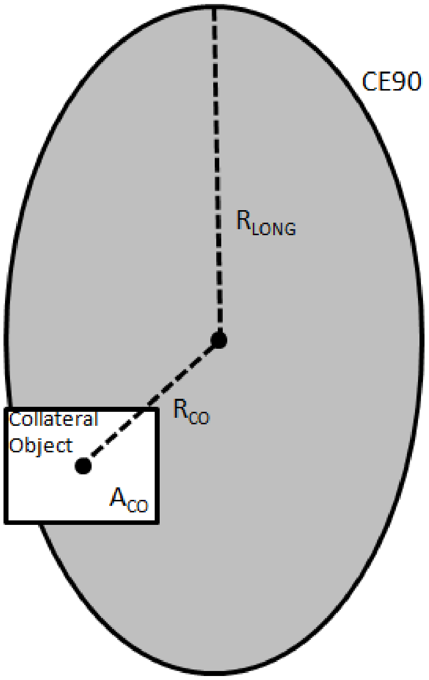


Figure 1: Current Collateral Damage Estimated Weighted Risk [99]

This calculation is performed for each collateral object and the results are aggregated:

$$\text{total collateral risk} \; = \frac{\sum A_W}{\text{total CE90 area}}. \tag{2}$$

While this equation presents a good start in estimation of collateral risk from airdrops, it does neglect a number of factors which affect proper estimation of collateral risk. Cammarano [23] improves this equation by finding the true distribution of airdrops, rather than simply the CE90. Based on operational drops, he argues that airdrops can be estimated using the bivariate normal distribution. Further, his estimation tool uses the bivariate normal distribution with the standard deviations in the x- and y-directions along with zero correlation between the x- and y-miss distances (i.e. $\rho = 0$).

Further, Cammarano provides for weights being given to the collateral objects to more accurately approximate real-world considerations. In the CDWRAT, the example provided treats bodies of water and buildings of all types and purposes as the same, with only the size and the distances to the center of the object taken into consideration. Cammarano's tool gives the planner the ability to say that while landing a bundle into a lake and into a occupied building are both undesirable, at least in the case of the lake, no one is injured, thus a higher weighting can be placed on the building. While Cammarano's work doesn't (typically) yield a percentage of collateral risk, it gives a much more meaningful statistic (overall collateral damage expected) to the decision-maker.

Cammarano also allows for buildings of differing shapes, where the CDWRAT only requires the center of the collateral object; this grants even more accuracy to his methodology. Cammarano also incorporates multiple bundle drops from the same mission, where each bundle has its own desired point of impact. Finally, Cammarano's tool gives a report for each collateral object providing accurate information to the decision-maker who might not want to assign weights to each of the collateral objects.

The second chapter of this dissertation leverages off of Cammarano's work. Once we are able to understand the nature of airdrops and estimate the collateral risk based upon relatively few inputs, the next question becomes: How do we then minimize the expected amount of collateral risk for an airdrop?

1.2.2 Pre-Planned Airstrike Collateral Damage. Pre-planned airstrikes are required to have a collateral damage estimation (CDE) done prior to engagement. The overall purposes of this CDE is to lower the amount of collateral damage resulting from the strike and to make the decision-maker fully aware of the collateral risks prior to making the decision to strike. The U.S. military has developed software which visually describes collateral risk in the area of the blast [43].

Much of the literature pertains to the blast effect of weapons on buildings, structures and people. Ngo et al. [77] and Mays and Smith [67] discuss blast effects on buildings, while Mills [71] and Newmark and Hansen [76] and a U.S. Army [101] report concern themselves with structural design to resist collateral damage. Humphrey et al. [49] discuss the effects of weapons on people within the blast and fragmentation radius.

Damage functions to model effects within the blast radius have been developed to accurately represent collateral risk in an airstrike. Driels [34], in his work on weaponeering, provides damage functions, and estimates of lethality on military targets. Douglass [33] presents a method for estimating collateral damage in urban environments based on the proximity of the friendly forces to the enemy combatants, based on size of weapons, likelihood of false alarm, blast radius, and circular error probable of weapon used. David [29] gave estimates of the safe distances in combat scenarios for friendly forces based on circular error radius, the lethal ranges of weapons, and the damage functions of these weapons. Lucas [64] tacked onto David's work by looking at the limiting behavior of damage functions relative to one another, focusing primarily on four damage functions: the lognormal, exponential, Gaussian, and cookie-cutter. Przemieniecki [82] discusses aspects of damage functions, accuracy functions, and collateral risk, as well as their effect on optimal aiming locations. Binninger [14] presents a lognormal damage function for use in predicting the effect of a nuclear weapon and by offsetting the aimpoint of the weapon can estimate the percentage of buildings destroyed at a given distance from the center of a town.

There has been minimal work in viewing the airstrike problem as a multi-criteria decision making model, most notably [57] and Brooks et al. [20] who used agent-based simulation to explore the trade-off between building damage and mission effectiveness. However, little has been done to develop the Pareto frontier within this framework to allow decision-makers the optimal aimpoints and employment methods to minimize risk while maximizing lethality on a military target.

1.2.3 Troops-in-Contact Collateral Damage. The majority of civilians killed in U.S. airstrikes died when Special Forces summoned an airstrike to support them during troops-in-contact (TICs) situations [39]. A TIC situation is an unplanned opportunity strike in support of ground forces that have made contact with enemy forces. In fact, only two (of 35) airstrikes resulting in civilian casualties in 2007-08 were from non-TIC (pre-planned) missions [38]. These rapid-response, fluid strikes are characterized by (typically) a lack of prior information concerning the nature and location of enemy and non-combatant forces, as well as friendly forces which may be in serious harm.

The ground forces in TIC situations, with the use of an Air Force JTAC, will request air support in order to strike the enemy, or at least provide them the opportunity to extricate themselves safely from the scene. The fog of war in the form of limited intelligence is the reason that TICs produce so many casualties. When friendly force lives are at stake, the air forces must act quickly and decisively, and the consequence of these actions may be the loss of Afghan civilian lives.

TICs scenarios have been the least studied of the three collateral hazards presented in this dissertation, yet they present the most danger to civilians. The closest information in the literature comes from other types of situations where quick decisions need to be made with limited information. Kocher et al. [56] look at the effects of time pressure on risky decisions and how pure loss and pure gain decision models affect human decision making. Decision-making where delaying the decision has an associated cost was studied by Payne et al. [79] where they found that in some cases delaying decision making results in a lower expected return even when the best decision is ultimately made.

Polikar [81] looks at decision-making where intelligence is gathered from multiple participants with differing perspectives, ultimately arguing that these varying perspectives yield better decisions. More germane to the battlefield, Phillips et al. [80] seek to model the flow of information in combat situations, a critical component of TICs. They present an information processing model which allows decision-makers to understand the best intelligence available.

The work presented here seeks to provide a framework for the issues and challenges which TICs present. The framework is a rough sketch of how information (the key issue in TICs) flows in fast-moving scenarios. Within this framework, we will seek to identify optimal decisions and optimal times to make these decisions during these rapid-response situations.

1.3 Methodology Literature Review

The three different problems presented within this dissertation yield varying formulations which touch on a variety of classes of operations research fields, including non-linear programming, global search techniques, evolutionary algorithms, multi-objective optimization, stochastic programming, and Markov decision processes. The developed formulations and solution techniques are explored in the following sections to give a background on which the three papers will be based.

1.3.1 Non-Linear Programming. Bazaraa et al. [11] formulate the non-linear minimization program as:

$$\begin{aligned} \min f(\mathbf{x}) & & \qquad (3)\\ \text{subject to } g_i(\mathbf{x}) &\leq 0 \text{ for } i = 1, \ldots, m \\ h_i(\mathbf{x}) &= 0 \text{ for } i = 1, \ldots, l \\ \mathbf{x} \in X, & \end{aligned}$$

where X is a subset of $\mathbb{R}^n$. f is a function from $\mathbb{R}^n \to \mathbb{R}$ and is referred to as the objective function. $g_j(\mathbf{x}) \leq 0$ and $h_k(\mathbf{x}) = 0$ are the constraints. In a non-linear program, the constraints and objective functions can be non-linear (whereas, in a linear program, all constraint and objective functions are linear).

If $f(\hat{\mathbf{x}}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$, then $\hat{\mathbf{x}}$ is the global minimum for the function f in the unconstrained problem. If there exists a neighborhood $N_\epsilon(\hat{\mathbf{x}})$ around $\hat{\mathbf{x}}$ where $f(\hat{\mathbf{x}}) \leq f(\mathbf{x})$ for any $\mathbf{x} \in N_\epsilon \hat{\mathbf{x}}$ then $\hat{\mathbf{x}}$ is a local minimum for f in $\mathbb{R}^n$.

If f is differentiable at $\hat{\mathbf{x}}$, and if $\hat{\mathbf{x}}$ is a local minimum, then $\nabla f(\hat{\mathbf{x}}) = 0$. Conversely, if f is differentiable at $\hat{\mathbf{x}}$ and there exists a vector $\mathbf{d}$ such that $\nabla f(\hat{\mathbf{x}})^t \mathbf{d} < 0$, then there exists a $\delta > 0$ such that $f(\hat{\mathbf{x}} + \lambda \mathbf{d}) < f(\mathbf{x})$ for each $\lambda \in (0, \delta)$ ($\mathbf{d}$ is the descent direction of f at $\hat{\mathbf{x}}$). The descent direction $\mathbf{d}$ represents a direction of improvement in an optimization problem that is used in most non-linear programming optimization algorithms, such as response surface methodology [74].

Similarly, in the constrained problem, if $f(\hat{\mathbf{x}}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in S$ where S is the feasible region for the problem, then $\hat{\mathbf{x}}$ is the global minimum and if there exists a neighborhood $N_\epsilon(\hat{\mathbf{x}})$ around $\hat{\mathbf{x}}$ where $f(\hat{\mathbf{x}}) \leq f(\mathbf{x})$ for any $\mathbf{x} \in N_\epsilon \hat{\mathbf{x}}$ then $\hat{\mathbf{x}}$ is a local minimum for f in S. With S as a non-empty set in $\mathbb{R}^n$ and $\hat{\mathbf{x}} \in \bar{S}$ then the cone of feasible directions (D) of S at $\hat{\mathbf{x}}$ is

$$D = \{\mathbf{d} : \mathbf{d} \neq \mathbf{0}, \text{ and } \bar{\mathbf{x}} + \lambda \mathbf{d} \in S \text{ for all } \lambda \in (0, \delta) \text{ for some } \delta > 0\}.$$

The cone of improving directions F at $\bar{\mathbf{x}}$ of f is

$$F = \{\mathbf{d} : f(\bar{\mathbf{x}} + \lambda \mathbf{d}) < f(\bar{\mathbf{x}}) \text{ for all } \lambda \in (0, \delta) \text{ for some } \delta > 0\}.$$

If f is differentiable at $\bar{\mathbf{x}} \in S$ then $\bar{\mathbf{x}}$ is a local optimum only if $F \cap D = \emptyset$. That is, there exists no feasible, improving direction for f at $\bar{\mathbf{x}}$. The concepts of feasible and improving directions are integral to area search methods over continuous (particularly, differentiable) objective functions.

While Lasdon [61] presents algorithms and heuristics for global optimization of large systems, Roy et al. [86] review commercially available packages for spreadsheet optimization. Achetti and Schoen [6] in their survey of global optimization techniques lump approaches into space-covering techniques, trajectory techniques (such as response surface methodology) and random search techniques, finding that each technique has its merit and no strategy is optimal without *a priori* information on the objective function.

1.3.1.1 Random Search Techniques. A random search technique is proposed by Solis and Wets [93] to find global minima in optimization problems expanding on the work of Anderson [4], Rastrigin [84], and Karnopp [53]. Their work is particularly useful in situations where function characteristics are difficult to compute, when the response function is "bumpy", when processing time is limited, and when it is highly desirable to find a global minimum among a large number of local minima. The assumption for the response function is that it is continuous, since a discontinuous function could conceivably have a minimum at a discontinuous point, which would be (nearly) impossible to find without an exhaustive search of every point in the input space S. Thus, they search for the essential infimum α of f on S which is defined as $\alpha = \inf\{t : v[x \in S | f(x) < t] > 0\}$, which is the set of points that yield values close to the essential infimum α has non-zero v-measure, meaning that the search is for a location where a set of points have a response less than t and this set also must have an interior (consider the case where the global minimum is at a discontinuous point x, for some $t \geq f(x)$ and $r > 0$ there exists no neighborhood $B(x; r)$ such that all points in $B(x; r)$ have a response value less than t).

Importantly, Solis and Wets note that any global search method must meet the assumption that for any Borel subset A of S with $v(A) > 0$, we have that $\prod_{k=0}^{\infty}[1 - \mu_k(A)] = 0$. In essence, this means that any subset A (with volume) of the search space S must be searched to guarantee that the global minimum is found.

The Solis and Wets algorithm uses normally-distributed steps to generate new points, the response value of the point is calculated and if the newly generated point has a higher (worse) objective function value, then steps are taken from the initial point in the opposite direction to find a new point. If both of these new points are worse than the original point then a new starting point is generated. Hart [44] notes that the Solis and Wets algorithm lacks definitive stopping criteria that yield optimality, typically relying on a fixed number of iterations. Additionally, Hart states, "In general, methods that utilize *a priori* information about a problem will outperform general purpose methods that utilize less information [44]."

Niederreiter [78] presents quasi-Monte Carlo methods for generating a sequence of uniformly distributed random points spread on a space. Estimates, using the variance of these random points, can be made for the value of the minima over the searched area and local search methods can be used in conjunction with these quasi-Monte Carlo techniques; however, global minimization again cannot be guaranteed on an objective function and domain without *a priori* information.

1.3.1.2 Response Surface Methodology.

Anderson [4] discusses experimental design and response surface methods to find input parameters for optimal performance in an uncharacterized experiment. Brooks [21] compares steepest ascent and univariate iterative methods for determining the optimal settings during experimentation, finding the ascent methods superior in terms of accuracy and speed.

A basic approach for approximating response functions is proposed by Myers and Montgomery [74] with $y = f(\xi_1, \xi_2, \ldots, \xi_k) + \epsilon$ where f is the true response function, which is either unknown or complicated. ϵ in the function for this work will represent sources of variation that are not accounted for by the derived model. $\xi_1, \xi_2, \ldots, \xi_k$ will be the input values for our model; in the airdrop model these will typically be the aimpoint (x and y) and the approach angle.

Myers and Montgomery discuss further the sequential nature of response surface methodology whereby initially hypotheses regarding the important input variables takes place, often backed up with a screening experiment. The screening experiment will identify the variables affecting the response variable and which variables' effects should be investigated further. After the screening takes place, they recommend the use of a first-order model and the method of steepest ascent, whereby starting from an initially small portion (referred to by Myers and Montgomery as the region of interest) of the overall search space, we begin to move in the direction of the optimal combination of input variables. Iteratively this method of steepest ascent is performed until a maximum for the response function is found (once the current solution can no longer be improved in the local area (region of interest)). Critically, it should be noted that the maximum found by this technique is simply a local maximum for the response function and is not guaranteed to be a global maximum. Situations arise where the response surface will be "bumpy" and have many local maxima throughout the total search space; thus while the techniques of Myers and Montgomery will be used, they must be extended in the effort to find a global maximum.

1.3.2 Evolutionary Algorithms. Hart [44] inspected genetic algorithms in combination with local search algorithms for solving global optimization problems. Michalewicz and Schoenauer [70] discuss adapting evolutionary algorithms to constrained parameter optimization problems, pointing out that finding a general algorithm that is optimal for all non-linear programs is unrealistic. The existence of local (and not global) optima presents the primary problem in non-linear programs with continuous functions, since steepest descent algorithms will yield only local (and not necessarily global) optima. Michalewicz and Schoenauer break down evolutionary algorithms into mutation operators, such as [10], [78] and [37], and crossover operators, as in [87], [35], [30] and later [94]. Mutation operators typically use Gaussian mutation to modify components of a solution vector, whereas crossover operators use multiple parent solution vectors to develop future generations of solution vectors. In both cases, the algorithms remove less-optimal solutions in each proceeding generation. Further, [70] views the constraint-handling methods as falling into four categories: feasibility preserving, penalty-based, feasibility/infeasibility separated, and hybrid methods.

Storn and Price [94] developed the differential evolution technique for solving global optimization problems. Their algorithm, discussed in much more detail in the following chapter, inspired a great deal of effort in the realm of optimization. Lampinen and Zelinka [59] apply the differential evolution algorithm to mixed integer-discrete-continuous problem demonstrating the versatility of the algorithm. Similar to what [70] presented for the general non-linear programming algorithm, Lampinen [58] presents a constraint-handling approach for the constrained differential evolution algorithm. Huang et al. [48] demonstrate a self-adaptive algorithm for constrained non-linear problems which modifies the two control parameters (F and CR) of differential evolution, thus cutting out the need for exhaustive parameter fine-tuning.

1.3.3 Multi-Objective Optimization.

Multi-objective optimization is applied in cases where there is more than one objective function.

$$\begin{aligned} \min y = f(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x})) \qquad (4) \\ \text{subject to } \mathbf{x} &= (x_1, x_2, \ldots, x_m) \in X \\ \mathbf{y} &= (y_1, y_2, \ldots, y_n) \in Y \end{aligned}$$

where $\mathbf{x}$ is the solution vector, X is the solution space, $\mathbf{y}$ is the objective vector, and Y is the objective space [107]. With more than one objective function, there becomes no strict ordering in the objective space (unlike in a single-objective problem). Therefore, the Pareto frontier concept is implemented, where we say that a solution $\hat{\mathbf{x}}$ is non-dominated if there exists no other $\mathbf{x}$ such that $f_k(\mathbf{x}) \leq f_k(\hat{\mathbf{x}})$ for all $k = 1, \ldots, n$ and $f_k(\mathbf{x}) < f_k(\hat{\mathbf{x}})$ for some $k \in 1, \ldots, n$. The collection of such non-dominated (Pareto-optimal) points forms the Pareto frontier for the formulation.

Techniques to solve multi-objective formulations attempt to find solutions near the Pareto frontier. However, since the Pareto-frontier is not a single point (typically) nor a finite collection of points (typically for continuous solution spaces), then finding solutions near the true Pareto frontier is not enough. Multi-objective algorithms also seek to find a variety of solutions that describe the Pareto-frontier more completely. Ziztler et al. [106] provide three metrics which describe the performance of multi-objective algorithms; these metrics are based on accuracy of the solutions, diversity of solutions, and breadth of solutions in creating the Pareto-frontier.

Evolutionary algorithms have been found to be particularly robust in developing the Pareto frontier for multi-objective problems due to their ability to process a set of solutions in parallel, therefore exploiting similarities of solutions by recombination [107]. Ziztler and Thiele [107] developed a strength Pareto evolutionary algorithm approach (SPEA) which stored nondominated solutions externally in a second group, evaluated solution fitness dependent on the number of external non-dominated points which dominate it, and clustered the nondominated point set in order to lower the nondominated set's population without losing diversity.

Babu and Jehan [9] implemented the work of Storn and Price [94] into the multi-objective realm by iteratively expelling dominated solutions for each generation. Xue et al. [104] present another differential evolution based algorithm (MODE) where non-dominated solutions are identified at every generation with the mutation step being different for non-dominated and dominated points. A Pareto-frontier differential evolution (PDE) is presented by Abbass et al. [2] which is seen to improve upon the SPEA approach of [107], with their approach constantly finding new non-dominated points in each generation and then removing similar ones based on a distance metric.

1.3.4 Dynamic Programming. Dynamic programming can be thought of using stages and states. A stage is a discrete point in time at which a decision (u_k) is made based on the state (x_k) of the system. The state is the summary of all decisions from previous stages and their outcomes, due not only to the decisions made but also the randomness (w_k) involved with moving from stage to stage. Some additive reward is gained for each decision made, and the goal is to maximize the sum of the rewards over the time horizon of the problem.

Bertsekas [13] lays out the main ingredients of a basic dynamic programming formulation as:

1. A discrete-time system of the form $x_{k+1} = f_k(x_k, u_k, w_k)$
2. Independent random parameters
3. A control constraint (decision)
4. An additive cost of the form $Eg_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k)$
5. Optimization over policies (rules for choosing u_k for each k and each possible value for x_k).

Denardo [31] formulates the dynamic programming problem as:

$$x_{k+1} = f_k(x_k, u_k, w_k), k = 0, 1, \ldots, N-1 \tag{5}$$

where:
k indexes discrete time,
x_k is the state of the system and summarizes past information that is relevant for future optimization
u_k is the control or decision variable to be selected at time k
w_k is a random parameter
N is the number of time control is applied.
For the LLS problem, we assume that each stage brings about more data about

the problem assuming the pilot continues to loiter. The rewards in this model are typically assumed to be costs, either the cost of a “look” decision or the likelihood of being incorrect given a “leave” or “shoot” decision. Ahner [3] applied approximate dynamic programming techniques to optimize control of unmanned aerial vehicles in combat situations.

1.3.5 Stochastic Programming.

Avriel and Williams [8] derive the expected value of information in recourse problems and show the value of a wait-and-see approach versus a recourse method. The difference between them is that in a recourse problem, a decision is made, then a random variable is observed, and then a recourse to a contingency plan is determined. A wait-and-see approach supposes that one could see what the random variable is before one makes an initial decision, and maximize our initial decision based on the known data rather than the unknown random variable. Certainly, the expected profit from the wait-and-see approach must be at least as great as in the recourse problem case. Then they pose a suggestion that if they could purchase the perfect information, how much should they pay for perfect information? $EVPI = WS - RP$ yields the expected value of perfect information. Avriel and Williams prove that $EVPI \geq 0$ given that an expected value of the random variable and the indicated maxima exist. Further, they show that $0 \leq EVPI \leq EV - RP$. EVPI can be applied to stochastic linear problems with recourse and more general stochastic programs including those with quadratic recourse.

1.3.5.1 Two-Stage Stochastic Linear Program.

Kall and Wallace [51] formulate the two-stage stochastic linear program as:

$$
\begin{aligned}
\text{min} \quad & c^T x + Q(x) \qquad (6)\\
\text{subject to} \quad & Ax = b, x \geq 0,\\
\text{where} \quad & Q(x) = \sum_j p^j Q(x, \xi^j)\\
\text{and } Q(x,\xi) \quad = \quad & \min\{q(\xi)^T y | W(\xi) y = h(\xi) - T(\xi)x, y \geq 0\},
\end{aligned}
$$

where p^j is the probability that $\widetilde{\xi} = \xi^j$, the jth realization of $\widetilde{\xi}$, $h(\xi) = h_0 + H\xi = h_0 + \sum_i h_i \xi_i$, $T(\xi) = T_0 + \sum_i T_i \xi_i$ and $q(\xi) = q_0 + \sum_i q_i \xi_i$.

Higle and Sen [46] present an algorithm for two-stage linear programs with recourse that leverages off of Benders' decomposition whereby they randomly generate observations of random variables to construct statisical estimates of supports of the objective function. Gassmann [40] presents a computer code for the multistage stochastic linear programming problem that uses an implementation of a nested decomposition algorithm.

Interior point methods are also considered by Birge and Holmes [16], Lustig, et al. [65], and Dantzig and Madansky [28]. Birge and Qi [17] are credited with applying Karmarkar's [52] interior-point method to stochastic programming. They formulate the stochastic linear program with recourse (and discrete random variables) as:

$$
\begin{aligned}
\text{min} \quad & c^T x_0 + \mathcal{Q}(x_0) && (7)\\
\text{subject to} \quad & A_0 x_0 = b_0 \\
& x_0 \geq 0, \\
\text{where} \quad & \mathcal{Q}(x_0) = \sum_{i=1}^{N} p_i Q(x_0, \xi^i) \\
\text{and for each} \quad & i = 1, \ldots, N, \\
\text{the recourse cost} \quad & \mathcal{Q}(x_0, \xi^i) \text{ is obtained by solving the recourse problem:} \\
& \mathcal{Q}(x_0, \xi^i) = \inf\{q^i y | W y = h^i - T^i x, y \in \mathbb{R}^{n_i+}\}, \\
& \xi^i = (q^i, h^i, T^i), \\
& p_i = \text{ prob } [\xi(\omega) = \xi^i].
\end{aligned}
$$

Birge and Qi noted that the dual block angular linear programs have the form:

$$
\begin{aligned}
\text{min} \quad & c^T x_0 + \sum_{i=1}^{N} c_i^T x_i && (8)\\
\text{subject to} \quad & A_0 x_0 = b_0 \\
& A_i x_0 + W_i x_i = b_i, i = 1, \ldots, N, \\
& x_i \geq 0, i = 0, \ldots, N \\
\text{where} \quad & x_i \in \mathbb{R}^{n_i}, i = 0, \ldots, N, \\
& b_i \in \ \mathbb{R}^{m_i}, i = 0, \ldots, N, \\
\text{where} \quad & m_i \leq n_i, i = 0, \ldots, N \\
\text{and} \quad & A_0, W_i \text{ have full row rank.}
\end{aligned}
$$

By substituting the expressions for $\mathcal{Q}$ into the stochastic formulation a linear programming formulation is created with $W = W_i$, $T^i = A_i$, and $p_i q^i = c_i$ for $i = 1, \ldots, N$. The resulting problem has $n = n_0 + N n_1$ variables and $m = m_0 + N m_1$

constraints. As Birge and Qi point out, methods for solving the linear formulation include Van Slyke and Wet's L-shaped method [102], Dantzig and Mandansky's decomposition method [28], and the basis factorization method proposed by Strazicky [95]. The L-shaped method solves the primal problem, while the decomposition and factorization methods solve the dual formulation.

Lustig et al. [65] base their work on scenario analysis, where a few realizations of the stochastic parameters are representative of the space of possible parameter outcomes. For a two-stage model, the size of the optimization problem grows linearly and typically, due to the size, decomposition methods are used. They point out that interior point methods make solving these large resulting models feasible. They implemented a primal-dual interior point method similar to the one described earlier, in which they show that the primal-dual method performs significantly better than Birge and Qi's [17] dual block angular approach. Additionally, they propose a partial splitting method which, due to the sparsity of the A matrices, speeds up the interior point methods considerably.

Birge and Holmes [16] formulated a dual affine algorithm starting with a dual feasible interior point, noting that the vast majority of the computational effort required is to calculate a solution to the symmetric positive definite system $(AD^2A^T)dy = b)$ or to calculate a factorization of the matrix that will enable quick solution of the system. Further, Blomvall and Lindberg [18] present Riccati-based primal interior point solver for multistage stochastic programming.

Birge and Holmes [15] also present a paper on the motivation for use of interior point methods for solving two-stage stochastic linear programs with fixed recourse along with characteristics of interior point solving methods. Additionally, they note that the size of stochastic linear programs can become extremely large due to the number of permutations of the unknown variables, and thus they present methods for speeding up the interior point methods, including reformulation of the program, transpose product factorization, and factorization of the dual block angular programs.

1.3.6 Markov Decision Process. A Markov decision process (MDP) is a problem in which there is a decision maker, a finite number of policies or choices the decision maker can choose, a transition probability matrix which defines the likelihood of the next state given the current state and policy, a transition reward matrix which indicates the current reward gained for the state and policy, and a performance metric based on the rewards gained during the stages of the MDP [42].

S, a finite state space of possible system states. A realization of the random variable S is denoted by s.
A, a finite set of actions. A realization of the random variable A is denoted by a. An action a causes transitions from the current state to some new state.
$T : S \times A \times S \to \mathbb{R}_{[0,1]}$ is the state-transition function, giving the probability that the agent transit to state s' when it is in state s and takes action a. In other words, the transitions specify how each of the actions and exogenous events change the state of the world. We denote by $T(s, a, s') = P(s'|a, s)$ this probability. We have for each s, $\sum_{s'} P(s'|a, s) = 1$.
$R : S \times A \to \mathbb{R}$ is the reward function giving the expected immediate reward gained by the agent for taking each action in each state.

Markov decision process applied to patient throughput in hospitals was researched by Broyles [22], while Qiu and Pedram [83] looked at the Markov decision process for continuous-time decision-making.

1.3.7 Partially-Observable Markov Decision Process. When the agent is unsure of the state s that he is currently in, unlike the MDP where the agent knows where he is at all times, this problem becomes a partially-observable Markov decision process (POMDP). There is some probability distribution around the state in which the agent thinks he is in (the belief state b). McAllister [68] looked at optimal planning with imperfect information, such as what U.S. troops have on battlefields.

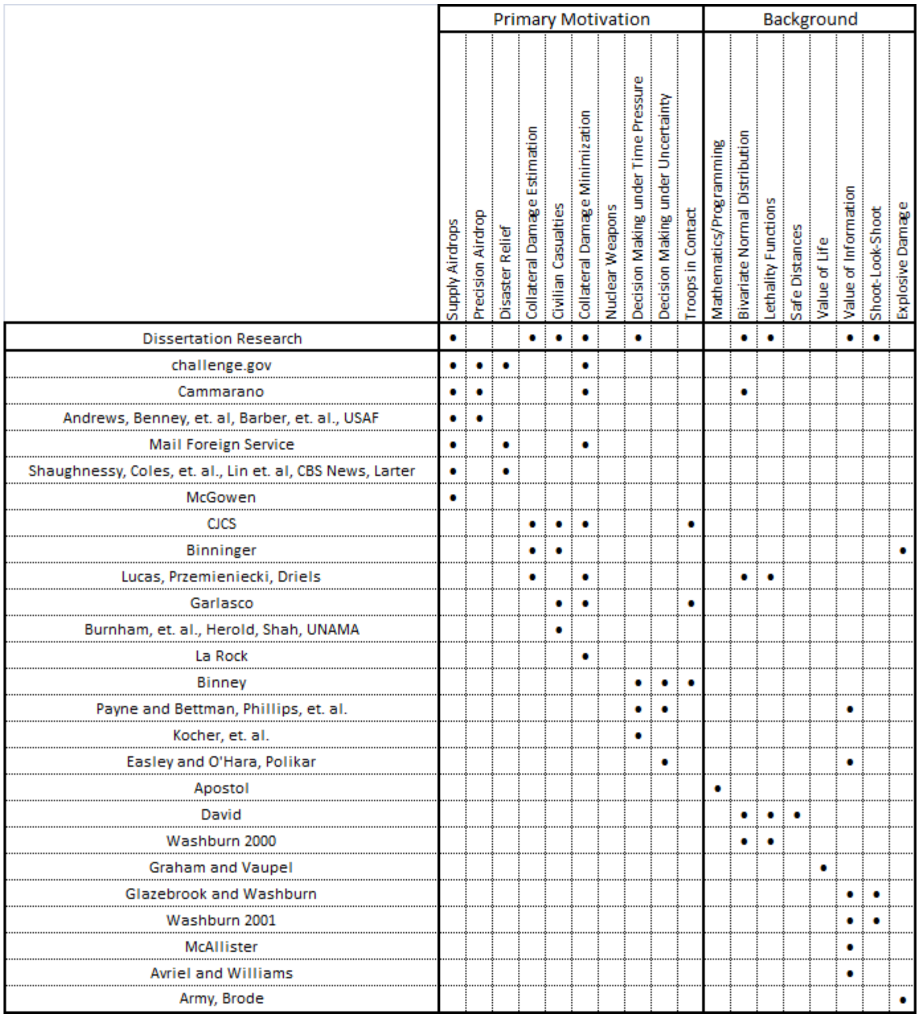
$O : S \times A \to \Pi(\Omega)$ is the observation function, which gives, for each action and resulting state, a probability distribution over all possible observations (we write $O(s', a, o)$ for the probability of making observation o given that the agent took action a and landed in state s').

Monahan [72] introduces a system where one of three decisions may be made. Either the observer can "inspect" - attempt to observe the true state of the target another time (at a cost), "stop" - make a determination as to the true state of the target and have no option for further observation, or "continue" in which he moves to the next time interval (at a cost) where the same three options will again be available to him. In the next time interval there is some probability that the nature of the target has changed which is a difference from the assumptions in this paper. Monahan concluded, "While the Markovian property does not hold for the state of the system, it does hold for the belief state of the system. The optimal policy for any given stage is only dependent on the current belief state and not decisions made in previous stages."

Monahan [73] later looked at the applications and theory behind partially observable MDPs. Kaelbling et al. [50] and Smallwood and Sondik [92] looked at optimal decision policies in partially observable MDPs. Yost and Washburn [105] applied linear programming techniques for decision making within POMDPs.

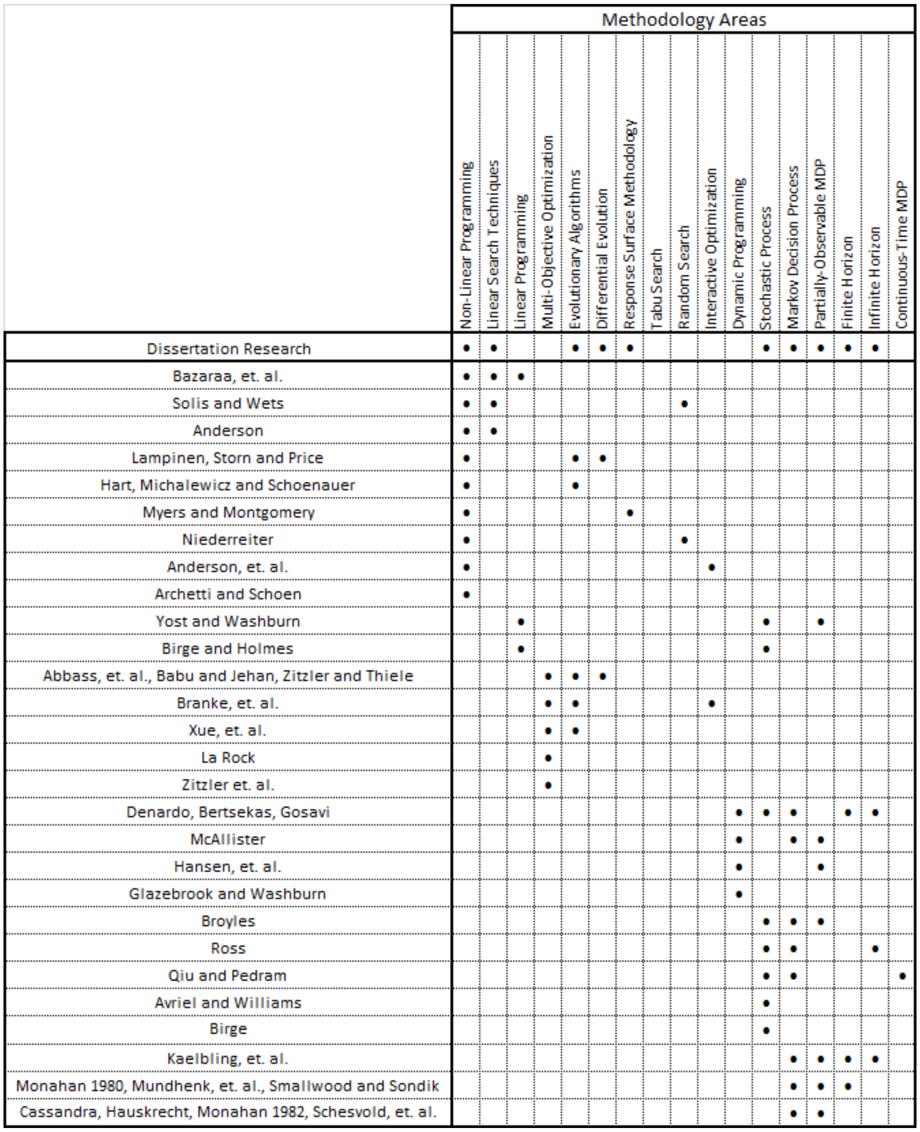
1.4 Overview of Literature Review

The following figures provide an overview of the topics covered and methodologies implemented in the following chapters.



	Primary Motivation										Background							
	Supply Airdrops	Precision Airdrop	Disaster Relief	Collateral Damage Estimation	Civilian Casualties	Collateral Damage Minimization	Nuclear Weapons	Decision Making under Time Pressure	Decision Making under Uncertainty	Troops in Contact	Mathematics/Programming	Bivariate Normal Distribution	Lethality Functions	Safe Distances	Value of Life	Value of Information	Shoot-Look-Shoot	Explosive Damage
Dissertation Research	•			•	•	•		•				•	•			•	•	
challenge.gov	•	•	•			•												
Cammarano	•	•				•						•						
Andrews, Benney, et. al, Barber, et. al., USAF	•	•																
Mail Foreign Service	•		•			•												
Shaughnessy, Coles, et. al., Lin et. al, CBS News, Larter	•		•															
McGowen	•																	
CJCS				•	•	•				•								
Binninger				•	•													•
Lucas, Przemieniecki, Driels				•		•						•	•					
Garlasco					•	•				•								
Burnham, et. al., Herold, Shah, UNAMA					•													
La Rock						•												
Binney								•	•	•								
Payne and Bettman, Phillips, et. al.								•	•							•		
Kocher, et. al.								•										
Easley and O'Hara, Polikar									•							•		
Apostol											•							
David												•	•	•				
Washburn 2000												•	•					
Graham and Vaupel															•			
Glazebrook and Washburn																•	•	
Washburn 2001																•	•	
McAllister																•		
Avriel and Williams																•		
Army, Brode																		•

Figure 2: Motivation and Background Literature Review Summary



	Methodology Areas																
	Non-Linear Programming	Linear Search Techniques	Linear Programming	Multi-Objective Optimization	Evolutionary Algorithms	Differential Evolution	Response Surface Methodology	Tabu Search	Random Search	Interactive Optimization	Dynamic Programming	Stochastic Process	Markov Decision Process	Partially-Observable MDP	Finite Horizon	Infinite Horizon	Continuous-Time MDP
Dissertation Research	•	•			•	•	•					•	•	•	•	•	
Bazaraa, et. al.	•	•	•														
Solis and Wets	•	•							•								
Anderson	•	•															
Lampinen, Storn and Price	•				•	•											
Hart, Michalewicz and Schoenauer	•				•												
Myers and Montgomery	•						•										
Niederreiter	•								•								
Anderson, et. al.	•									•							
Archetti and Schoen	•																
Yost and Washburn			•									•		•			
Birge and Holmes			•									•					
Abbass, et. al., Babu and Jehan, Zitzler and Thiele				•	•	•											
Branke, et. al.				•	•					•							
Xue, et. al.				•	•												
La Rock				•													
Zitzler et. al.				•													
Denardo, Bertsekas, Gosavi											•	•	•		•	•	
McAllister											•		•	•			
Hansen, et. al.											•			•			
Glazebrook and Washburn											•						
Broyles												•	•	•			
Ross												•	•			•	
Qiu and Pedram												•	•				•
Avriel and Williams												•					
Birge												•					
Kaelbling, et. al.													•	•	•	•	
Monahan 1980, Mundhenk, et. al., Smallwood and Sondik													•	•	•		
Cassandra, Hauskrecht, Monahan 1982, Schesvold, et. al.													•	•			

Figure 3: Methodology Literature Review Summary

1.5 Description of Research

This dissertation first seeks to understand the nature of both airdrops and airstrikes in terms of the parameters and distributions that accurately represent all aspects of these missions. Once the parameters are understood, a formulation for each of these problems is sought. Optimization techniques and algorithms will then be created to minimize collateral risk while adhering to mission and logistical constraints. Finally, sensitivity analysis along with lessons-learned will be presented to provide take-aways for mission planners acting in these environments under strict time and mission requirements.

1.6 Statement of Original Contribution

This dissertation seeks to fill in gaps in both the literature and the methodology by which the USAF estimates and minimizes collateral risk. The major contributions presented in the following three chapters are:

- Characterization of airdrop distribution based on real-world data.
- Formulation of the collateral damage problem.
- Comparison of non-linear programming algorithms for solving the airdrop collateral damage minimization problem.
- Algorithm for quickly finding optimal airdrop parameters based on a surrogate function for the bivariate normal distribution.
- Multi-objective formulation for the airstrike collateral damage problem.
- Algorithm for finding Pareto optimal solutions for the airstrike problem.
- Quantitative comparison of damage functions for use in estimating collateral risk.
- Quantitative comparison of weapon employment guidelines in the collateral airstrike problem.
- Formulation of limited intelligence airstrike problem.
- Quantitative comparison of effects of weighting and *a priori* intelligence on optimal firing policy.

II. Minimizing Supply Airdrop Collateral Damage Risk

2.1 Background

2.1.1 Introduction. Supply airdrops occur for a variety of reasons. Supplies are airdropped to scientists at the South Pole, Humvees to American troops at forward operating bases in the mountains of Afghanistan, and food and water to Haitians in the days after their devastating 2010 earthquake.

The necessity of airdrops as part of emergency disaster relief is underscored in [91], [27], [62]. Shortly after the 2010 Haitian earthquakes, for example, United States Air Force (USAF) planes were dropping over 200 water and food bundles per day outside Port-au-Prince [24] from 100 daily flights [66]. Bottlenecks on the roads in Haiti along with the blockage of the seaport prevented the movement of critical supplies, forcing the primary source of aid to be airdrops into secured areas [66]. The airdropped supplies helped minimize widespread violence and looting in the days following the earthquake.

Supply airdrops are typically used in cases where plane landings are either unsafe or inefficient. In the first four months of 2011, the USAF dropped 25 million pounds of supplies for troops and locals in Afghanistan and Iraq. This was not possible by truck. Supply airdrops constitute a vital tactical piece of both war-fighting and peacekeeping missions for the USAF throughout the world and consequently the USAF has developed expertise potentially useful to other supply airdrop agencies. Airdrops allow ground units to operate in areas that are not tied to ground logistical resupply. Aerial resupply allows the freedom of movement without worrying about convoys and their large logistical footprint. [60]

Supply airdrops have risks beyond those of the equipment and personnel involved. A recent challenge.gov request [26] underscored the danger that comes with dropping humanitarian food and water supplies over populated areas (where they may be in highest demand) and the need to develop alternative methods of performing such drops. With the uncertain flight paths of airdrops, along with their weight (up to thousands of pounds), airdrops are particularly dangerous ventures when occurring even in sparsely-populated areas. Poorly planned or executed airdrops can result in lost, ruined, or stolen cargo and, more importantly, collateral damage to the people and buildings near the drop zone. This is compounded by the fact that an airdrop is typically not a single object - rather a series of objects referred to as a bundle. This article develops a new technique for estimating risk of collateral damage associated with supply airdrops and an efficient method for finding the optimal aimpoint and approach direction for supply missions so as to minimize collateral damage. We demonstrate, based on real-world drop data, that only an estimate of the standard deviations in the x- and y- directions, with respect to flight path, is required to estimate the expected risk of collateral damage during a supply mission. The standard deviation parameters fit a bivariate normal distribution that characterizes the error of a drop. Risk of damage is estimated by integrating the bivariate normal distribution over the areas of undesirable landing locations in the drop zone for each object in the supply bundle dropped.

Once an estimate of collateral damage risk is established, the goal becomes to find the aimpoint and flight approach angle which minimize collateral damage yet result in a drop as close to the recipients as possible. We must also accommodate the reality that different elements in the scene may have different values of avoidance (e.g. an occupied building versus a lake). The nature of this search is highly non-linear because:

- of the shape of the bivariate normal distribution,
- each object in the bundle has its own drop error distribution, and
- each element in the scene has its unique location, shape, size, and value.

To develop an effective global search technique for this problem, response surface methodology (RSM), differential evolution (DE), and random search (RS) methods are compared and combined in this paper to provide quick and effective tools for finding this optimal solution. The best search algorithm will be shown to be orders of magnitude faster than enumeration and up to 20% more accurate than the use of maps and the naked eye.

2.1.2 Nature of Airdrops. Airdrop accuracy has been an ever-present challenge to airdrop supply planners since the early days of resupply via aircraft airdrop. Techniques such as high-velocity airdrops for rugged cargo minimize the effects of wind on airdrop trajectory and maintain accuracy while allowing for higher release altitudes and increased aircraft survivability. “Reefing” is an airdrop beginning descent at high velocity for target accuracy and then switching to low velocity in mid-descent. This allows aircraft to drop cargo from higher altitudes with the accuracy of a lower altitude drop. Many of these techniques and technologies were born out of operational necessity and can be used in combination with different chute and aircraft types.

One of the most successful recent examples of accuracy improvement is the Joint Precision Airdrop System (JPADS). JPADS uses a steerable parachute and an airborne guidance unit to control the cargo’s descent and guide it to its desired point of impact [69]. JPADS offers many advantages over traditional airdrops: increased accuracy, reduced drop zone size requirements, standoff cargo release, improved aircraft survivability, and immediate feedback on airdrop accuracy [12]. A disadvantage of JPADS is its cost relative to traditional “dumb” airdrops (which, by the way, comprise the majority of supply airdrops). In order to keep costs down recovering and reusing retrograde airdrop items is necessary, though not always feasible [12]. The challenge is that an agency providing supply airdrop support may not have the budget or access to the best techniques and may need to do the best they can with the equipment they have. This is a main motivation for the development of our methodology.

Regardless of drop technology used, planners must choose carefully where to target. If a drop is too far from the point of use, recovery personnel could be exposed to hazard and delay in getting relief to the recipients. If it is too close to ground personnel or collateral objects, then the consequences of cargo weighing several tons traveling at speeds of over 50 feet per second are unacceptable. How do mission planners know how close is too close? What is the chance that the cargo will impact a collateral object inside the drop zone?

Airdrop errors occur when an airdrop does not land at its intended point of impact. These errors are commonly described as a distance from the drop target and an angle with respect to the drop zone axis or by (x, y) coordinates. These errors can be caused by problems with the computed air release point, flight path error, drop crew error, drop zone elevations, cargo ballistics, load weight, or unpredicted winds. While the calculation of a release point takes into account many factors (summarized graphically in Figure 4) to determine the correct location in the air to release an airdrop from the aircraft, individual drops are always subject to drop error. In the next section, we characterize those errors probabilistically.

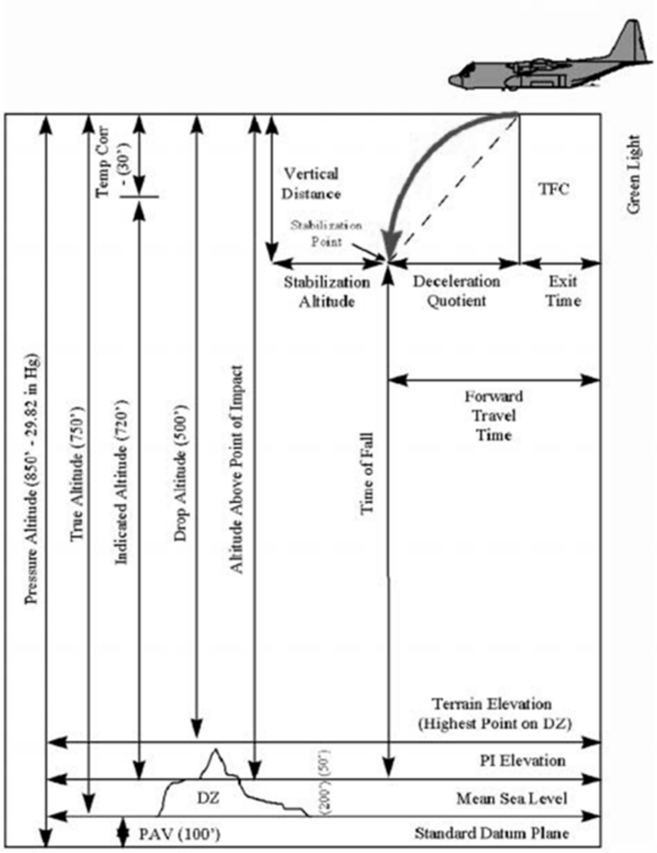


Figure 4: Drop Zone Planning Diagram [99]

2.1.3 Bivariate Normal Distribution of Airdropped Bundles. We find that, under a wide range of drop conditions and technologies, the drop errors from supply bundles fit the bivariate normal distribution. The data set we studied was provided by the USAF Air Mobility Command (AMC). It is actual (as opposed to practice run) data from over 700 airdrops in the field. Figure 5 shows a plot of all of the data. It is not GPS data. You can almost imagine the ground spotter radioing that a particular drop was "50 meters long at your 2 o'clock". The data set has unique characteristics which we studied at length. For more detail on that analysis, see [23]. Our main findings follow.

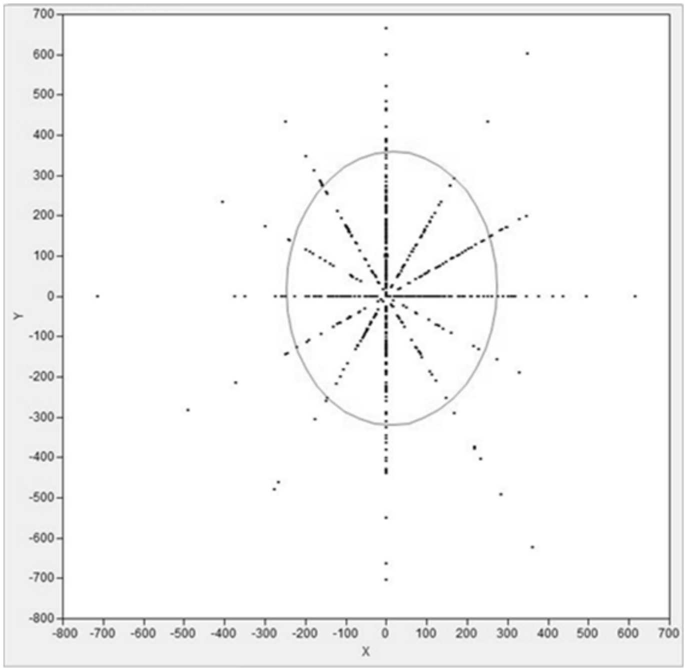
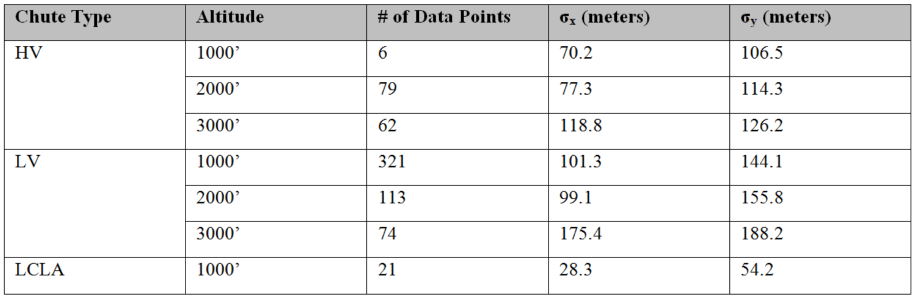


Figure 5: Airdrop Scatter

We find that the mean errors in both the x and y direction are statistically zero. On average (and this result remained when we parse the data into different technologies and conditions), the planners hit where they aimed. However, there is substantial variation. Further, the x and y directional errors have different standard deviations. This makes sense because typically the timing of an airdrop affects the y direction whereas wind has the majority of the effect on the x miss distance. We also find that drop errors in the x-dimension are uncorrelated from those in the y-dimension (i.e. $\rho = 0$) which makes risk calculations using the bivariate normal distribution simpler, but if $\rho \neq 0$ then the same algorithms would be used; run time would simply be longer.

It is worth stopping here to consider the implication of these findings to an airdrop planner. The only data necessary to characterize the errors in a supply drop are the standard deviations in the x and y directions for the equipment being used in a particular drop zone. This can be accomplished with relatively few data points after making relatively few flights. Practice drops could even be done away from the drop site to assess accuracy.

As an example, we can characterize the types of supply airdrops that AMC made in our data set. The standard deviations for combinations of chute type and airdrop altitude are summarized in Figure 6, along with the number of data points collected for each combination. We find that the chute type and the airdrop altitude have a statistically significant effect on the error distribution patterns, but not the aircraft type. An AMC planner merely looks up a value pair from the right two columns of the figure to characterize fully the shape of the risks of their drops.



Chute Type	Altitude	# of Data Points	σ_x (meters)	σ_y (meters)
HV	1000’	6	70.2	106.5
	2000’	79	77.3	114.3
	3000’	62	118.8	126.2
LV	1000’	321	101.3	144.1
	2000’	113	99.1	155.8
	3000’	74	175.4	188.2
LCLA	1000’	21	28.3	54.2

Figure 6: Standard Deviation Table

To give the reader a sense of what the risk profiles look like, Figure 7 depicts the density function for seven individual objects in a bundle airdrop mission. The distance (five units) between the elements is found by multiplying the aircraft speed at drop by the time interval between releases. Figure 7 shows the effect of increasing standard deviation. When the standard deviation of the drop objects is low, their individual probability distributions can be easily identified as multiple modes. On the other hand, with a small separation distance relative to the standard deviations, the graph becomes smoother and the bundle drop error profile approaches unimodality.

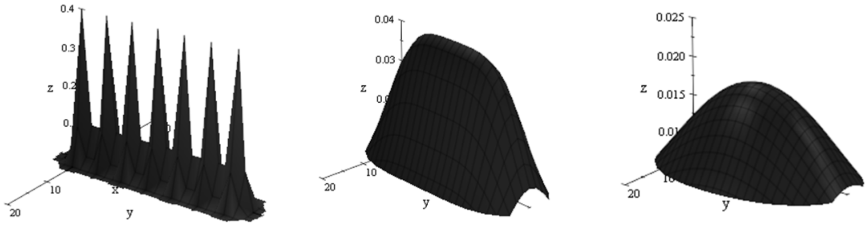


Figure 7: Example Multiple Bivariate Normal Distributions
$(d = 5, n = 7, \sigma_x = \sigma_y = 1, 5, 10)$

One of these bundle shapes, dropped into the landing scene, is how we characterize collateral damage risk. Numerically, we integrate the compound bivariate normal distribution over the undesirable landing areas. In the next section, we begin the development of an optimal location algorithm based on bivariate bundle risks.

2.2 Classes of Applicable Global Search Algorithms

It is not obvious how to design an optimum-seeking algorithm for this problem. It is one of the contributions of this paper. In this section, we introduce several important candidate global search algorithms.

2.2.1 Random Search. Random search technique is proposed by Solis and Wets [93] to find global minima in optimization problems expanding on the work of Anderson [4], Rastrigin [84] and Karnopp [53]. The Solis and Wets algorithm uses normally distributed steps to generate new points, the response value of the point is calculated and if the newly generated point has a higher (worse) objective function value, then steps are taken from the initial point in the opposite direction to find a new point. If both of these new points are worse than the original point then a new starting point is generated. Hart [44] notes that the Solis and Wets algorithm lacks definitive stopping criteria that yield optimality, typically relying on a fixed number of iterations. However, their work is particularly useful in situations where function characteristics are difficult to compute, when the response function is "bumpy", when processing time is limited, or when it is highly desirable to find a global minimum among a large number of local minima. The assumption for the response function is that it is continuous, since a discontinuous function could conceivably have a minimum at a discontinuous point, which would be (nearly) impossible to find without an exhaustive search of every point in the input space. All of these characteristics are exactly the conditions of the collateral damage problem assuming bivariate normally distributed bundle drops.

Niederreiter [78] presents quasi-Monte Carlo methods for generating a sequence of uniformly distributed random points spread on a space. Estimates, using the variance of these random points, can be made for the value of the minima over the searched area and local search methods can be used in conjunction with these quasi-Monte Carlo techniques, however, global minimization cannot be guaranteed on an objective function and domain without *a priori* information. Hart [44] notes importantly: "In general, methods that utilize *a priori* information about a problem will outperform general purpose methods that utilize less information." For example, a method to specifically find minima for the bivariate normal problem could use random search, but take advantage of solving a more specific problem than a general search or general algorithm is made to solve.

2.2.2 Response Surface Methodology. Another important basic approach for approximating response functions is proposed by Myers and Montgomery [74] with $y = f(\xi_1, \xi_2, \ldots, \xi_k) + \epsilon$ where f is the true response function, which is either unknown or complicated. ϵ in the function for this work will represent sources of variation that are not accounted for by the derived model. $\xi_1, \xi_2, \ldots, \xi_k$ will be the input values for our model; in the airdrop model these are the aimpoint (x, y) and the approach angle.

Myers and Montgomery further discuss the sequential nature of response surface methodology whereby initial hypotheses regarding the important input variables take place, often backed up by a screening experiment. The screening experiment identifies the variables affecting the response variable and which variables' effects should be investigated further. After screening takes place, they recommend the use of a first-order model and the method of steepest descent, where starting from an initially small portion (referred to by Myers and Montgomery as the region of interest) of the overall search space, the user begins to move in the direction of the optimal combination of input variables. Iteratively this method of steepest descent is performed until a minimum for the response function is found in the local region of interest. It should be noted that the minimum found by this technique is simply a local minimum for the response function and is not guaranteed to be global.

In each local area of the drop scene, we need to find a local minimum if we desire to obtain the global minimum. Therefore, we will consider the response surface methodology of Myers and Montgomery. Specifically, when the random search produces top candidates we will use RSM to improve the local solutions.

2.2.3 Differential Evolution. Storn and Price [94] present the differential evolution heuristic for global optimization over continuous spaces, sometimes referred to generically as genetic algorithms. Differential evolution does not rely on the cost function to be differentiable or even continuous. The airdrop problem presents a continuous cost function, but one where the differentiation of the cost functions has no closed-form solution. Differential evolution is a parallel direct search method which utilizes D-dimensional (in the airdrop problem, 3-dimensional) parameter vectors as a population for each generation G. The initial group of vectors is chosen randomly and will cover the entire parameter space. DE generates new parameter vectors by adding the weighted difference between two population vectors to a third vector in a process called mutation. The mutated vector is then mixed with another predetermined "target" vector to yield the "trial" vector. The objective function value of the “trial” vector is compared to that of the “target” vector. In the selection step, whichever vector has the lower function value will be the “target” vector for the next generation. Each generation of vectors makes improvement in the cost function value, and the mutations help prevent getting trapped in local minima. Additionally, keeping DE vectors for each generation prevents trapping in the local minima. The baseline model of Storn and Price is *DE/rand/1/bin* meaning that the initial vectors are randomly chosen, there is one difference vector, and the crossover scheme is binary distributed.

For constrained differential evolution, constraints are dealt with by the insertion of a boundary penalty into the objective function [94]. Michalewicz and Schoenauer [70] note that the methods for dealing with constraints in a genetic algorithm can be handled in four ways:

- methods which preserve feasibility of the solutions,
- methods which use penalty functions, such as Storn and Price [94],
- methods which make distinctions between feasible and infeasible solutions, and
- hybrid methods.

Lampinen [58] discusses the laborious and difficult nature of the selection of penalty parameters, and proposes a method that either preserves feasibility (if the previous generation’s solution was feasible), moves towards feasibility (if both the current and previous generations’ solutions are infeasible), or moves towards optimality (if both generations’ solutions are feasible). The Lampinen approach doesn’t rely upon starting solutions which are feasible. For the use of differential evolution in the airdrop problem disallowing solutions that fall outside of the feasible region would also be acceptable since the differential evolution method will not select these disallowed solutions in the “selection” step of the algorithm. Michalewicz and Schoenauer would classify this approach as a method which makes “distinctions between feasible and infeasible solutions.”

This approach appears to have merit. The airdrop problem does not have overly complex constraints that preclude the easy generation of multiple, feasible starting solutions, and the objective function is not differential - both properties well served by evolutionary approaches.

2.3 Methodology

2.3.1 Formulation of Optimal Supply Airdrop Location. Assuming a bivariate normal distribution with known parameters σ_x, σ_y, and ρ, our formulation is:

$$\min_{\hat{\theta},\hat{x},\hat{y}} \sum_{j=1}^{m} v_j (1 - \prod_{i=1}^{n} (1 - \int_{x_{j\,\min}}^{x_{j\,\max}} \int_{y_{j\,\min}}^{y_{j\,\max}} \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{\frac{-[\frac{(x-\hat{x}_i)^2}{\sigma_x^2} - 2\rho\frac{(x-\hat{x}_i)}{\sigma_x}\frac{(y-\hat{y}_i)}{\sigma_y} + \frac{(y-\hat{y}_i)^2}{\sigma_y^2}]}{2(1-p)^2}} dydx)) \tag{9}$$

subject to $\theta_{\min} \leq \hat{\theta} \leq \theta_{\max}$ (if desired)

$x_{\min} \leq \hat{x} \leq x_{\max}$

$y_{\min} \leq \hat{y} \leq y_{\max}$

where m is the number of collateral objects

$\hat{x}_i = \hat{x} + s(i-1)\sin\hat{\theta}$

$\hat{y}_i = \hat{y} + s(i-1)\cos\hat{\theta}$

v_j - value of the j^{th} collateral object

$\hat{x}_i$ - longitude of the aimpoint of the i^{th} bundle

$\hat{y}_i$ - latitude of the aimpoint of the i^{th} bundle

$\hat{\theta}$ - approach angle

σ_x - horizontal miss distance standard deviation

σ_y - vertical miss distance standard deviation

n - number of objects in the airdrop bundle

s - distance of separation of the objects in the airdrop

An important part of the formulation is that likelihoods of individual collateral objects being struck must be modified to account for the possibility of multiple hits. The overall probability of a hit is found by $1 - \prod_{i-1}^{n}(1 - p_i)$ where p_i is the likelihood of an individual airdropped object striking a specific collateral object. The value of the collateral objects - that is the value of avoiding them - can be set to any positive value; if they are to be treated equally then they are all set to 1. The solution to our formulation is the aimpoint and approach angle which minimizes the total collateral value of the bundles striking collateral objects, which is different than choosing the aimpoint and angle which have the lowest likelihood of striking any collateral objects.

A final constraint is added to avoid unbounded solutions which occur anywhere outside the scene or potentially at the edge of the scene, because there is no collateral damage to be avoided there. Based on our experience, we allow solutions which produce only airdrops in which the middle of the bundle lands no closer than two grid lines from the boundary of the scene. Remember, however, this does not guarantee that all the objects in a bundle will actually land within the scene due to the uncertainty of the bundle flight paths.

In the following section we introduce a base problem, motivated by a real-world airdrop supply scene. It is intended to make concrete what we are doing and be the starting point to develop test problems for the specific algorithms we will develop and compare next.

2.3.2 A Drop Zone Problem Solved. Our drop zone is a sparsely populated setting. This is typical of a humanitarian supply drop area selected to be near a city, but not in the city. Figure 8 shows the scene we have defined with (shaded) elements to be avoided and the optimal drop location and angle found (the series of circles). Throughout this paper, the sizes of the elements shown in the drop scene are to scale, while the sizes of the circles are proportional (but not to scale) to the magnitude of the standard deviations of drop errors. Let us discuss setting up and solving this example in detail.

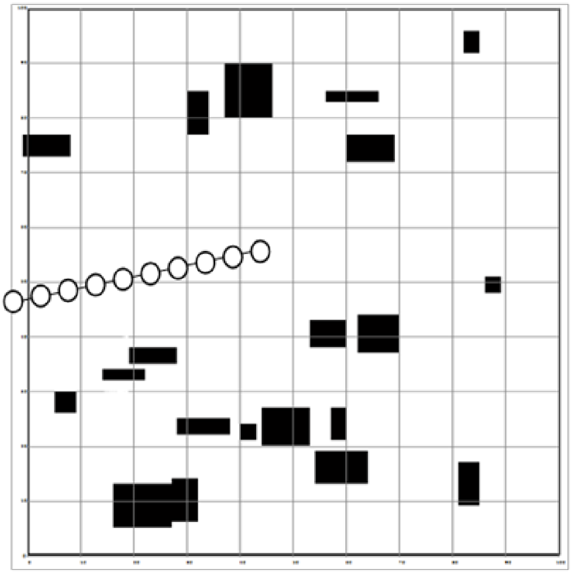
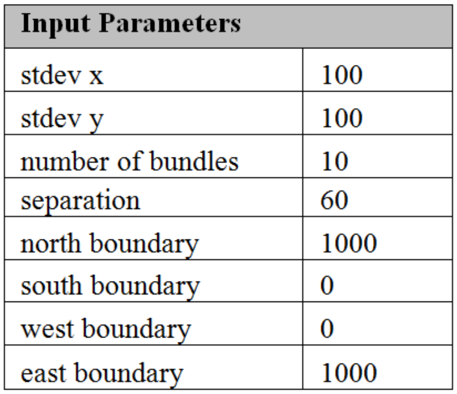


Figure 8: Scenario Layout with Optimal Aiming Location

We have chosen to characterize the search space as 1000 x 1000 meters with 100 (10 x 10) grid zones 100 m by 100 m, with the requirement that the middle of the bundle object lands at least 200 m from the edge. Keeping the size of the search space small is important, both because it determines the magnitude of the optimization problem and it bounds the area where the recovery team will have to travel to acquire the dropped supplies.

In this base problem, an airdrop plane traveling at 120 meters/second drops objects 0.5 seconds apart. This yields a distance between the bundle objects of 60 meters so that with ten objects dropped there is a total path length of 540 m. The drop technology involved has standard deviations of 100 m in both the x and y direction (typical values across the types of airplanes and chute types that AMC uses). Figure 9 collects this data together and is actually the format of an input screen for our solution program.

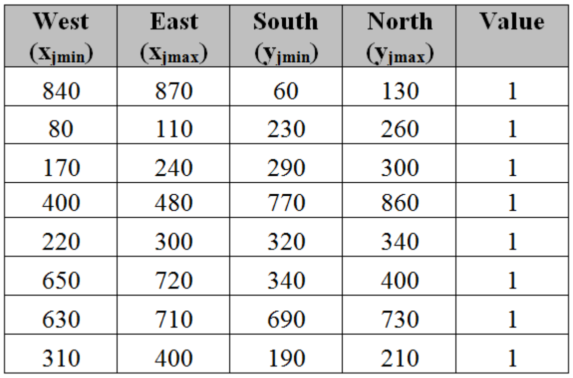


Input Parameters	
stdev x	100
stdev y	100
number of bundles	10
separation	60
north boundary	1000
south boundary	0
west boundary	0
east boundary	1000

Figure 9: Input Parameter Table

The collateral objects, and their avoidance values, are entered via another table. Figure 10 contains the data for our base problem with 20 collateral objects. For the purposes of this paper, collateral objects are taken to be rectangular facing the axes of the coordinate system. (As an aside, circular buildings are well represented by a square.) Complex shapes can be built with several rectangles. A larger object could in fact be a cluster of buildings. This scenario intentionally includes a variety of objects with lengths and widths between 10 and 100 meters. We are envisioning every collateral object as being occupied housing with equal avoidance value. Of course if an object is known to be just a barn, its value could be decreased.

Note that we have chosen to use a coordinate system of the cardinal directions. This is not required but it makes planning with maps, GPS, or satellite imagery easier and, regardless, the solution algorithm picks its optimal angle in terms of whatever coordinate system is selected.



West (x_{jmin})	East (x_{jmax})	South (y_{jmin})	North (y_{jmax})	Value
840	870	60	130	1
80	110	230	260	1
170	240	290	300	1
400	480	770	860	1
220	300	320	340	1
650	720	340	400	1
630	710	690	730	1
310	400	190	210	1

Figure 10: Collateral Objects

In terms of solution, collateral objects and their values determine the more attractive drop paths and locations. Figure 8 shows the true optimal solution in this scenario. Unsurprisingly, the optimal aiming location for the bundles lies in the rather large gap in the buildings on the western side of the layout. Dropping in this location yields a minimum objective function value of 0.088 which means that, in an individual bundle drop, an average of only 0.088 collateral objects will be struck (since each object has a value of 1) by the ten bundle objects. Note that there is a small chance that some of them may be struck more than once.

2.3.3 Solution Methods. In this section, we undertake a series of studies comparing, combining, and evaluating the global search methods of Section 2.2 to solve the formulation of Section 2.3.1 for problems of the type in Section 2.3.2.

2.3.3.1 Surrogate Functions. Regardless of the search approach used, calculations of the cost function are computationally expensive since there must be a complicated integral computed at each point on the grid that lies within a collateral object. Surrogate functions are routinely used in evolutionary algorithms when the cost function is complicated and requires a large amount of processing time. This is the case for the multiple integrations of the bivariate normal distribution for the airdrop collateral damage problem. To combat this, for the first number of generations of differential evolution we will use a surrogate function which requires less processing time. The key, however, is to find the correct number of generations at which to make the switch between using the surrogate cost function and the actual cost function. Using the surrogate cost function too long will result in convergence to a sub-optimal solution (a solution that is optimal for the surrogate function, but not the true cost function). Switching to the actual cost function too soon will negate the time savings gained by using the surrogate cost function.

The surrogate function created for the integration of the bivariate normal distribution will be based on an approximation of its probability density function (pdf). By experimentation, we find that just four rectangular prisms approximate this pdf adequately. Figure 11 shows graphically the normal distribution approximation used for the surrogate function.

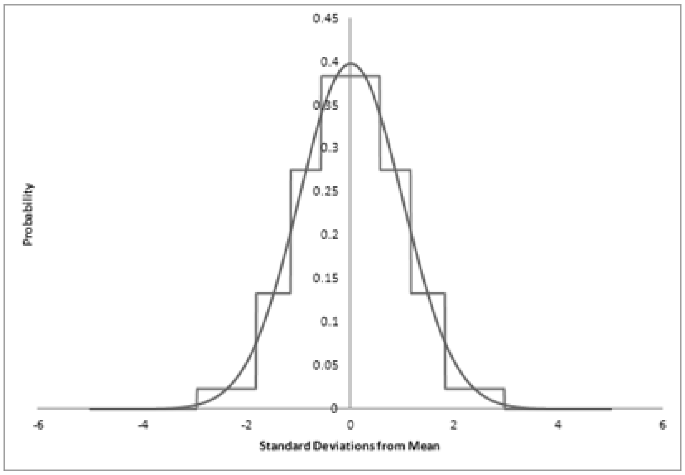


Figure 11: Surrogate Approximation of the Normal Distribution

As evidence of the accuracy of the 4-point surrogate, Figure 12 compares the objective function value of the surrogate normal to the actual probability value for over 10,000 bundle object and collateral object impacts. The graph shows low error and high correlation (0.894) between the two functions. The speed-up of the surrogate does not deter the accuracy of the search to optimum.

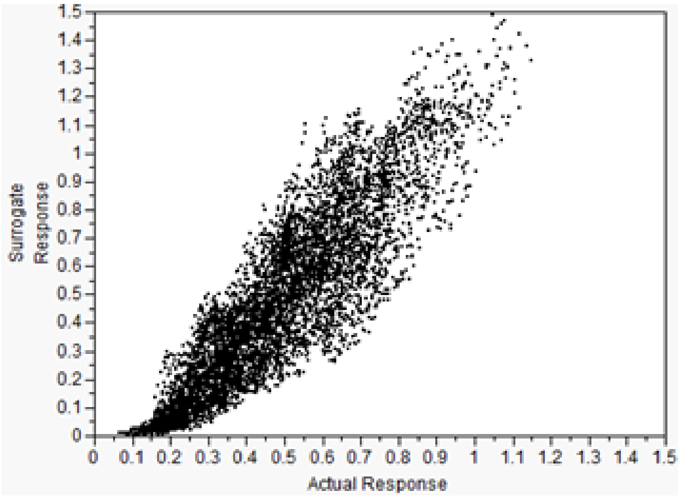
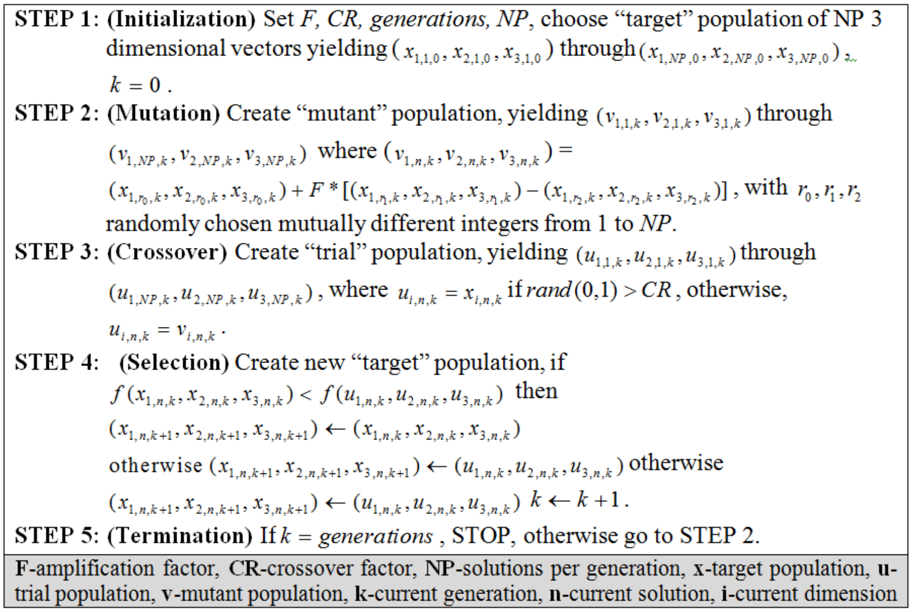


Figure 12: Surrogate versus Actual Objective Function

2.3.3.2 Differential Evolution Algorithm. We have implemented the differential evolution method described in Section 2.2.3 and Figure 13 created by Storn and Price [94]. The differential evolution algorithm has as its key inputs the number of generations and the number of solutions within the generations. After a series of trials we find the following constants for the differential algorithm provided convergence to one solution while keeping the run time for the algorithm low: F = 0.8, CR = 0.5, NP = 40, and generations = 100. This means there are 4,000 separate calculations of the objective function.



STEP 1: (Initialization) Set *F, CR, generations, NP*, choose "target" population of NP 3 dimensional vectors yielding $(x_{1,1,0}, x_{2,1,0}, x_{3,1,0})$ through $(x_{1,NP,0}, x_{2,NP,0}, x_{3,NP,0})$, $k = 0$.

STEP 2: (Mutation) Create "mutant" population, yielding $(v_{1,1,k}, v_{2,1,k}, v_{3,1,k})$ through $(v_{1,NP,k}, v_{2,NP,k}, v_{3,NP,k})$ where $(v_{1,n,k}, v_{2,n,k}, v_{3,n,k}) =$ $(x_{1,r_0,k}, x_{2,r_0,k}, x_{3,r_0,k}) + F * [(x_{1,r_1,k}, x_{2,r_1,k}, x_{3,r_1,k}) - (x_{1,r_2,k}, x_{2,r_2,k}, x_{3,r_2,k})]$, with r_0, r_1, r_2 randomly chosen mutually different integers from 1 to *NP*.

STEP 3: (Crossover) Create "trial" population, yielding $(u_{1,1,k}, u_{2,1,k}, u_{3,1,k})$ through $(u_{1,NP,k}, u_{2,NP,k}, u_{3,NP,k})$, where $u_{i,n,k} = x_{i,n,k}$ if $rand(0,1) > CR$, otherwise, $u_{i,n,k} = v_{i,n,k}$.

STEP 4: (Selection) Create new "target" population, if
$f(x_{1,n,k}, x_{2,n,k}, x_{3,n,k}) < f(u_{1,n,k}, u_{2,n,k}, u_{3,n,k})$ then
$(x_{1,n,k+1}, x_{2,n,k+1}, x_{3,n,k+1}) \leftarrow (x_{1,n,k}, x_{2,n,k}, x_{3,n,k})$
otherwise $(x_{1,n,k+1}, x_{2,n,k+1}, x_{3,n,k+1}) \leftarrow (u_{1,n,k}, u_{2,n,k}, u_{3,n,k})$ otherwise
$(x_{1,n,k+1}, x_{2,n,k+1}, x_{3,n,k+1}) \leftarrow (u_{1,n,k}, u_{2,n,k}, u_{3,n,k})$ $k \leftarrow k+1$.

STEP 5: (Termination) If $k = generations$, STOP, otherwise go to STEP 2.

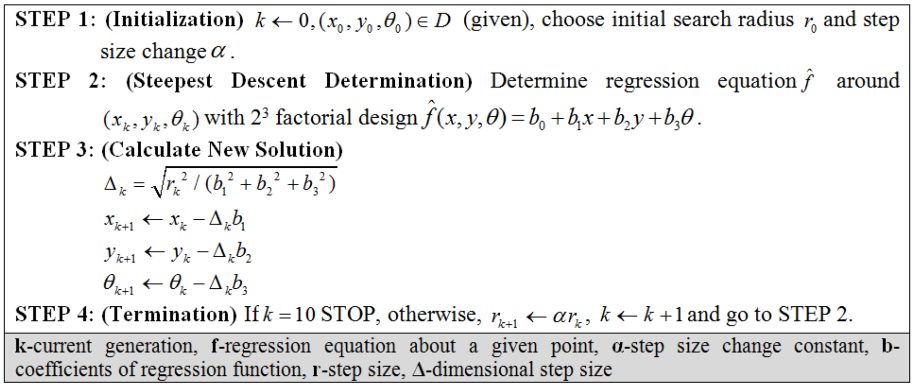
F-amplification factor, **CR**-crossover factor, **NP**-solutions per generation, **x**-target population, **u**-trial population, **v**-mutant population, **k**-current generation, **n**-current solution, **i**-current dimension

Figure 13: Differential Evolution Algorithm

2.3.3.3 Differential Evolution with Surrogate. We have implemented two variations of the Differential Evolution algorithm where the surrogate function is used to speed up the process. In the first variation, the initial 50% of the generations use the surrogate function, and in the second the surrogate function is used in 100% of the generations. Once the final generation of solutions is obtained, those locations have their true objective function values calculated and then compared in order to determine the best solution. Random starting points within the user-defined grid initialize both algorithms.

2.3.3.4 Response Surface Algorithm. For the response surface algorithm, the solution space is initially searched in order to find good candidate solutions. This step is accomplished by uniformly searching the solution space, finding the best ten solutions and then using these ten solutions as inputs to the second phase of the algorithm. The second stage takes each of these ten solutions and then determines the regression equation about each solution. From there (again, for each solution) movement is made in the direction of the steepest descent by a given step size. Those new ten solutions then become the inputs and the process is repeated with smaller step sizes, for a given number of iterations. Finally, the ten resulting solutions are compared to determine the best solution. For the response surface method in our base problem scenario, for example, there were 7*7*35 + 2*2*3*10*10 = 2915 separate calculations of the cost function computed.

Once the ten best solution vectors from enumeration are found, each solution is treated as the starting point and the algorithm in Figure 14 is performed:



```
STEP 1: (Initialization) $k \leftarrow 0, (x_0, y_0, \theta_0) \in D$ (given), choose initial search radius $r_0$ and step
        size change $\alpha$.
STEP 2: (Steepest Descent Determination) Determine regression equation $\hat{f}$ around
        $(x_k, y_k, \theta_k)$ with $2^3$ factorial design $\hat{f}(x, y, \theta) = b_0 + b_1 x + b_2 y + b_3 \theta$.
STEP 3: (Calculate New Solution)
        $\Delta_k = \sqrt{r_k^2 / (b_1^2 + b_2^2 + b_3^2)}$
        $x_{k+1} \leftarrow x_k - \Delta_k b_1$
        $y_{k+1} \leftarrow y_k - \Delta_k b_2$
        $\theta_{k+1} \leftarrow \theta_k - \Delta_k b_3$
STEP 4: (Termination) If $k = 10$ STOP, otherwise, $r_{k+1} \leftarrow \alpha r_k$, $k \leftarrow k + 1$ and go to STEP 2.
```

k-current generation, **f**-regression equation about a given point, **α**-step size change constant, **b**-coefficients of regression function, **r**-step size, **Δ**-dimensional step size

Figure 14: Response Surface Methodology Algorithm

2.3.3.5 Response Surface Methodology with Surrogate. We have also programmed the response surface algorithm with surrogate method using the surrogate objective function to perform the enumeration step before switching to the true objective function for the response surface portion. Thus, the best ten solutions from the surrogate objective function are found and on these solutions the response surface algorithm is performed (using the true objective function), giving movement towards better solutions moving from these ten solutions for a series of iterations. Examples of the search movement are shown in Figure 15. Starting points are at the grid intersections and the points are the midpoint of the bundle drop. This figure shows the unique nature of search evolution in our problem wherein the angle changes during the search and not merely the location.

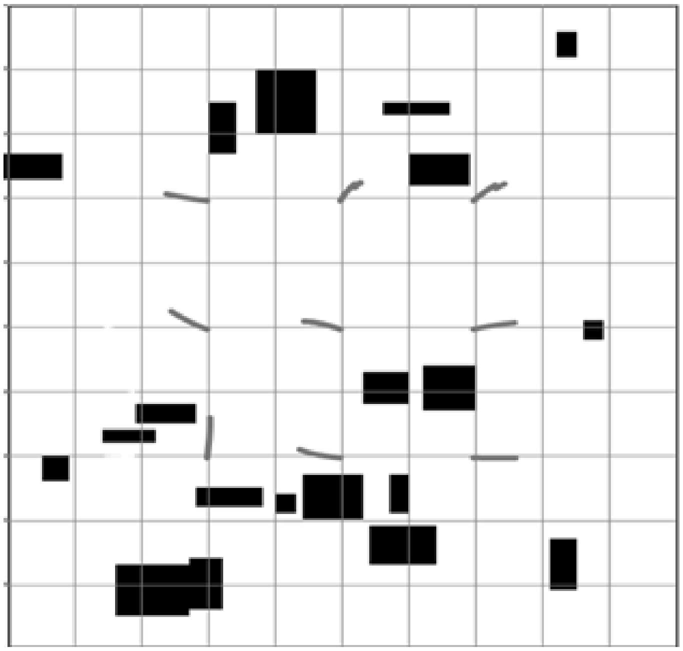


Figure 15: Steepest Descent Movement in RSM

2.3.3.6 Enumeration. One might ask, why not simply evaluate at many points in the drop area and pick the best one of those as the estimate of the global optimum? The problem is computational effort - when searching the entire solution space in an enumerative manner, the size of the steps taken has a significant effect on both the optimal solution found and the number of trials necessary to find the optimal solution. As the step size approaches zero, the number of trials approaches infinity at a rate inversely proportional to the cube of the step size (since this is a three-dimensional problem). Additionally, as the step size approaches zero, the solution found approaches the true optimal solution for the scenario. Figure 16 shows, in fact, this tradeoff in our base scenario problem. Enumeration can, in fact, find the optimal solution of 0.088. The problem is that, for our example, it requires 673,501 computations of the objective function.

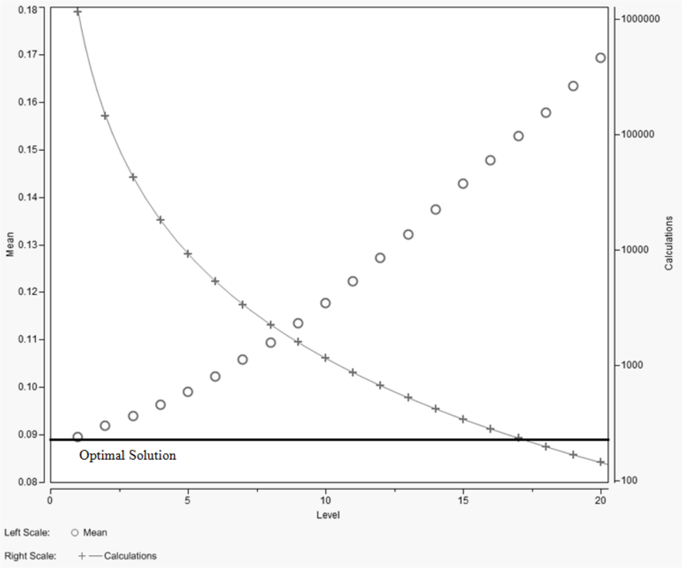


Figure 16: Optimal Result and Calculations vs. Step Size

Nevertheless, enumeration offers a comparison value in terms of accuracy and speed. We have gone so far as to implement enumeration with the surrogate technique. We will also evaluate the behavior of enumeration on a coarser grid.

2.4 Global Search Results

In this section, we compare all of the algorithms introduced in this paper. The merit metrics are running time and accuracy. The results are all solutions to the base scenario of Section 2.3.2; however our computational experience leads us to believe that the relative performance of the algorithms is the same on other problems of our type.

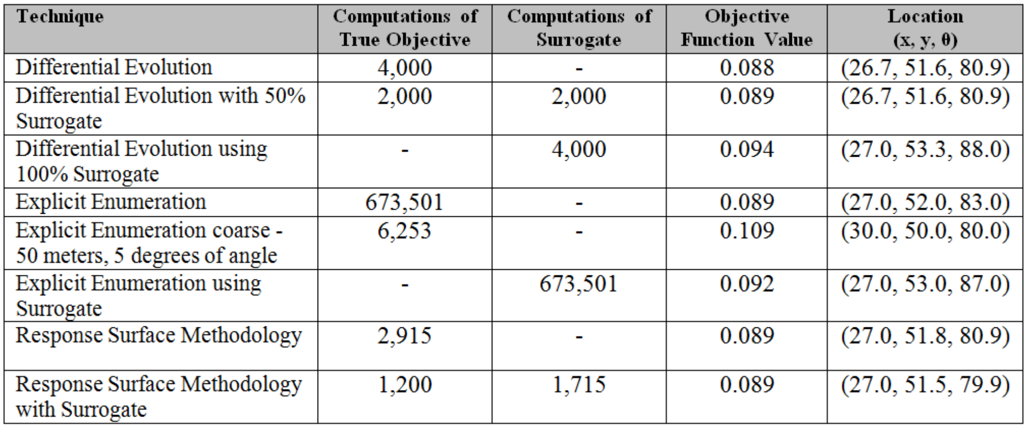
In addition, we investigate the behavior of solutions for a wide range of scenario variations that a planner might face. This serves the purpose of validating our work and, more importantly, reveals tradeoffs and improvement methods that we consistently find.

2.4.1 Comparison of Search Algorithms. Figure 17 collects the results of our algorithm comparison studies. As discussed earlier, there are three DE algorithms, three enumeration algorithms, and two RSM algorithms. Let us first look within the algorithm types and then across them.

The DE algorithm consistently finds (and should by its design) the best solution because it is defined in terms of the true objective function. Surrogacy speeds the calculation up, but the quality of the solution can suffer.

The enumeration algorithm benefits the most from surrogacy because of the high number of objective function evaluations required. Trying to improve computational time by using a coarser grid does not work as the solution quality inevitably suffers. This is the behavior we saw in Figure 16.

The RSM algorithm produces high quality results but has the limitation of always approximating the true objective function. This is compounded by the surrogacy approximation. With that said, RSM still produces good quality solutions consistently in a reasonable time.



Technique	Computations of True Objective	Computations of Surrogate	Objective Function Value	Location (x, y, θ)
Differential Evolution	4,000	-	0.088	(26.7, 51.6, 80.9)
Differential Evolution with 50% Surrogate	2,000	2,000	0.089	(26.7, 51.6, 80.9)
Differential Evolution using 100% Surrogate	-	4,000	0.094	(27.0, 53.3, 88.0)
Explicit Enumeration	673,501	-	0.089	(27.0, 52.0, 83.0)
Explicit Enumeration coarse - 50 meters, 5 degrees of angle	6,253	-	0.109	(30.0, 50.0, 80.0)
Explicit Enumeration using Surrogate	-	673,501	0.092	(27.0, 53.0, 87.0)
Response Surface Methodology	2,915	-	0.089	(27.0, 51.8, 80.9)
Response Surface Methodology with Surrogate	1,200	1,715	0.089	(27.0, 51.5, 79.9)

Figure 17: Summary of Results for Techniques

Figure 18 summarizes the computational cost versus solution accuracy of the algorithms. Computational cost is calculated as the number of calculations of the true objective function plus 25% of the number of calculations of the surrogate function (that being the approximate savings). The lower left corner of Figure 18 is the most desired having a low calculation time and a low optimal solution value. The computationally-intensive method of enumeration is dominated and is not the way to approach solving this class of problem. The remaining Pareto optimal approaches then are differential evolution, response surface method using the surrogate function, and the differential evolution method using only the surrogate function.

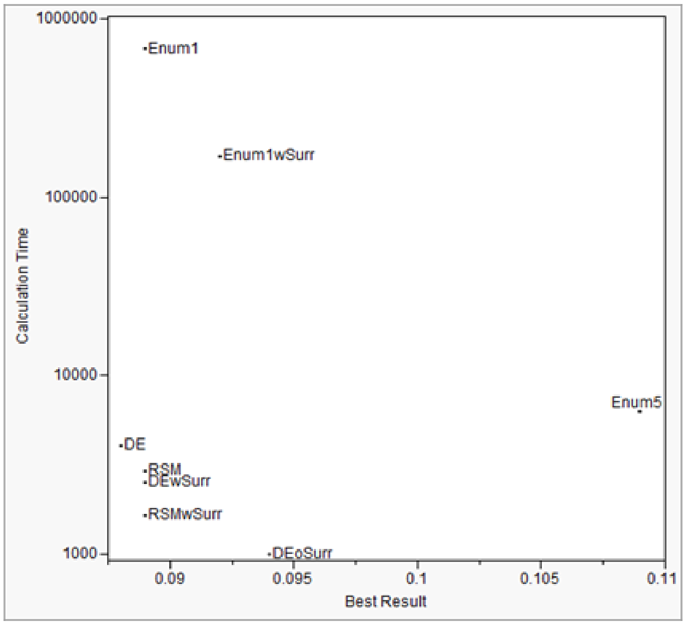


Figure 18: Results of Various Methods

We feel that, although it is fast, the DE with 100% surrogate gives away too much in terms of accuracy. Either of the other two Pareto methods is a good choice, depending on the tradeoff for the planner in terms of speed versus accuracy. For example, using a 2.7 GHz desktop PC with 4.0 GB RAM, we obtain solutions to the base problem in a few minutes of clock time. So, choosing pure DE might mean an 8 minute response time versus a 2 minute response for RSM with surrogate (whose solution might be 1-2% worse). For accuracy, and consistency of comparison, we use DE for all the remaining case studies in this paper.

2.4.2 Guidelines to Airdrop Planners. In this section, we solve fourteen variations of a new airdrop problem (Scenario #1 below) using the techniques developed in this paper to explore the effects of changing inputs (the drop technology, the scene itself, etc.). The variations represent real-world situations that an airdrop planner might face.

Choosing bundle configurations:

- Using bundles of size five rather than ten. This investigates the impact of two missions of five versus one mission with a size ten bundle (Scenario #2).
- Using a bundle separation distance of 30 meters rather than 60 meters. This shows the benefit of a plane traveling slower over the dropzone (#3).

Choosing drop technology (the standard deviation values were taken from Figure 6):

- Using higher (#4), lower (#5), and unequal standard deviations (#6) to generally underscore the effects of accuracy in delivery systems on collateral damage.
- Specifically using an LCLA chute at 1000 feet (the lowest standard deviations of all chute-altitude combinations) (#7).
- Specifically using an LV chute at 3000 feet (the highest standard deviations of all chute-altitude combinations) (#8).
- Specifically using an LV chute at 1000 feet (the most common chute-altitude combination) (#9).
- Specifically using an HV chute at 2000 feet (the most common HV altitude) (#10).

Effect of changes in the nature of collateral objects:

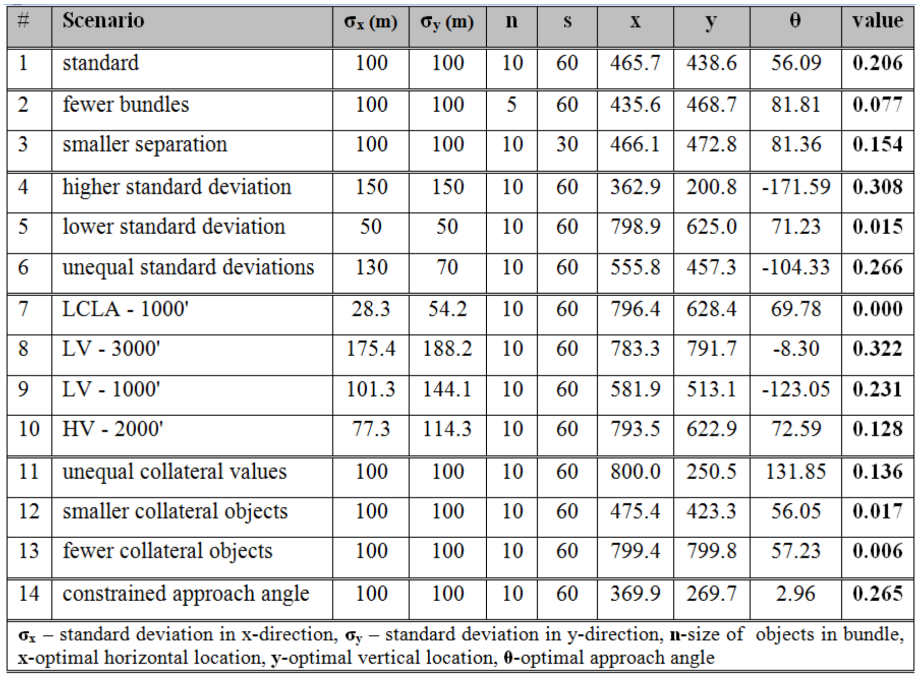
- Using differing values for the collateral objects rather than all collateral objects having the same values. Collateral objects are randomly given values between 0 and 2 rather than the previous common value of 1 (#11).

- Using smaller collateral objects ranging from 10 x 10 meters to only 30 x 30 meters rather than up to 100 x 100 meters. This demonstrates the benefit of more accurate intelligence on the nature of the collateral objects in the drop scene (#12).
- Using ten collateral objects in the scene rather than twenty demonstrating the benefit of moving to more sparsely populated areas (#13).

Consequences of limitations in travel over the drop scene:

- Using a flying angle constraint (this is the “as desired” constraint in our formulation). This demonstrates that weather, terrain, or other safety flying logistics can have detrimental consequences on drop risk. (#14)

The inputs and results for all scenarios are presented in Figure 19, wherein the last column is the damage value at optimum (lower is better). Like scenarios are grouped together.



#	Scenario	σ_x (m)	σ_y (m)	n	s	x	y	θ	value
1	standard	100	100	10	60	465.7	438.6	56.09	**0.206**
2	fewer bundles	100	100	5	60	435.6	468.7	81.81	**0.077**
3	smaller separation	100	100	10	30	466.1	472.8	81.36	**0.154**
4	higher standard deviation	150	150	10	60	362.9	200.8	-171.59	**0.308**
5	lower standard deviation	50	50	10	60	798.9	625.0	71.23	**0.015**
6	unequal standard deviations	130	70	10	60	555.8	457.3	-104.33	**0.266**
7	LCLA - 1000'	28.3	54.2	10	60	796.4	628.4	69.78	**0.000**
8	LV - 3000'	175.4	188.2	10	60	783.3	791.7	-8.30	**0.322**
9	LV - 1000'	101.3	144.1	10	60	581.9	513.1	-123.05	**0.231**
10	HV - 2000'	77.3	114.3	10	60	793.5	622.9	72.59	**0.128**
11	unequal collateral values	100	100	10	60	800.0	250.5	131.85	**0.136**
12	smaller collateral objects	100	100	10	60	475.4	423.3	56.05	**0.017**
13	fewer collateral objects	100	100	10	60	799.4	799.8	57.23	**0.006**
14	constrained approach angle	100	100	10	60	369.9	269.7	2.96	**0.265**

σ_x – standard deviation in x-direction, **σ_y** – standard deviation in y-direction, **n**-size of objects in bundle, **x**-optimal horizontal location, **y**-optimal vertical location, **θ**-optimal approach angle

Figure 19: Summary of Results for Scenarios

Figure 19 demonstrates that cutting the standard deviation in half from 100 meters (#1) to 50 meters (#5) lowers the expected amount of collateral damage risk by 93%, underscoring the need for accurate delivery systems. The vast improvement possible by a technologically-advanced delivery system can be seen in the near-zero damage caused when using the LCLA chute type from 1000 feet (#7). Conversely, the LV chute type from 3000' (#8), with its large standard deviations has a collateral damage risk 61% higher than the standard case (#1).

Figure 19 also shows the sharp decline in expected damage from lowering the number of collateral objects (#13) or decreasing the size of collateral objects (#12). This is to be expected (and quantified by our model). It is more surprising, that a 50% decrease in the separation (#3) from flying slower yields only a small decrease in expected collateral damage (25%).

Let us turn from the optimum values of the objective function to the changes in the location and angle of drop. Figures 20 through 26 depict the optimal aiming locations and angles from Figure 19 for all fourteen scenarios:

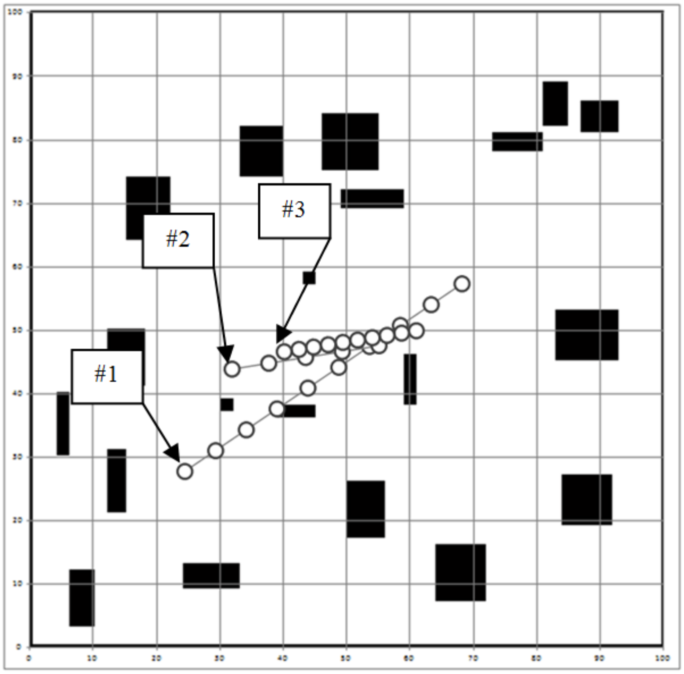


Figure 20: Varying Bundles - #1, 2, 3

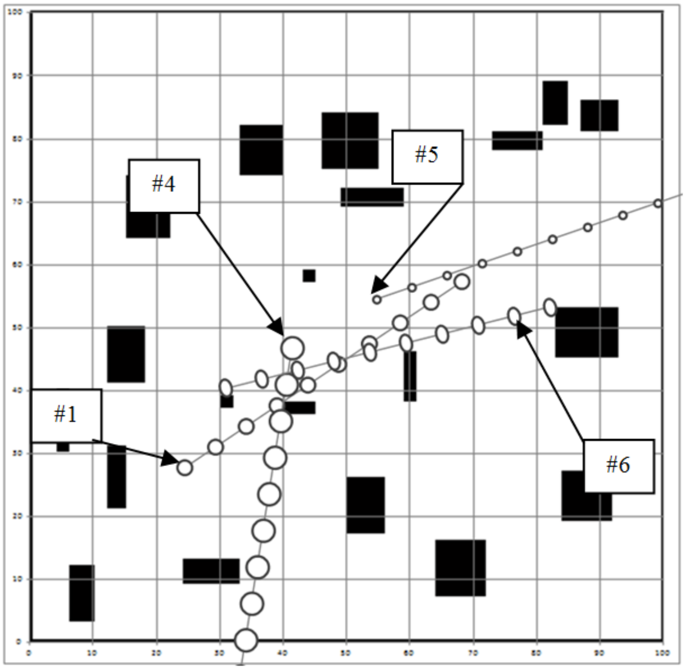


Figure 21: Varying Standard Deviation - #1, 4, 5, 6

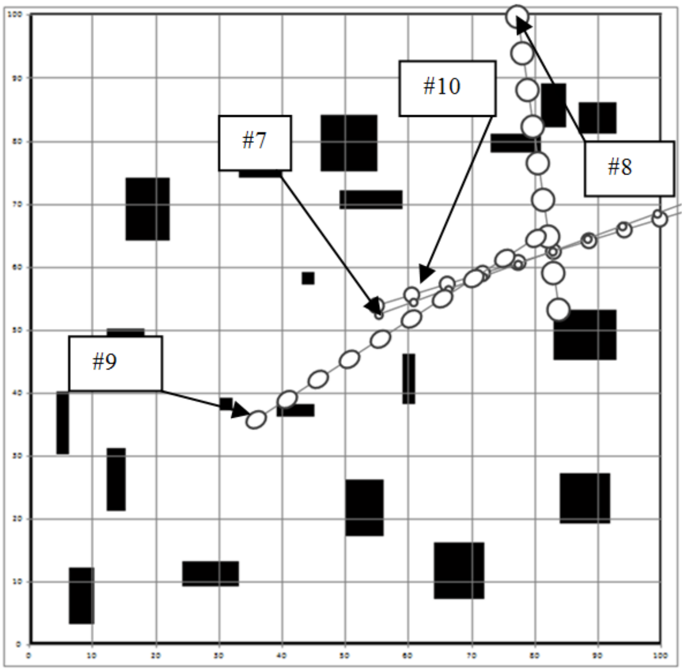


Figure 22: Chute Type - Altitude - #7, 8, 9, 10

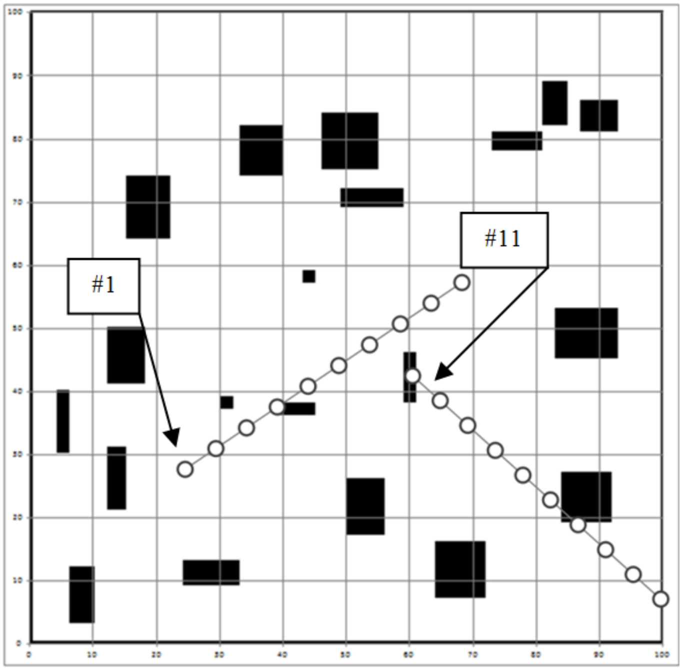


Figure 23: Varying Collateral Values - #1, 11

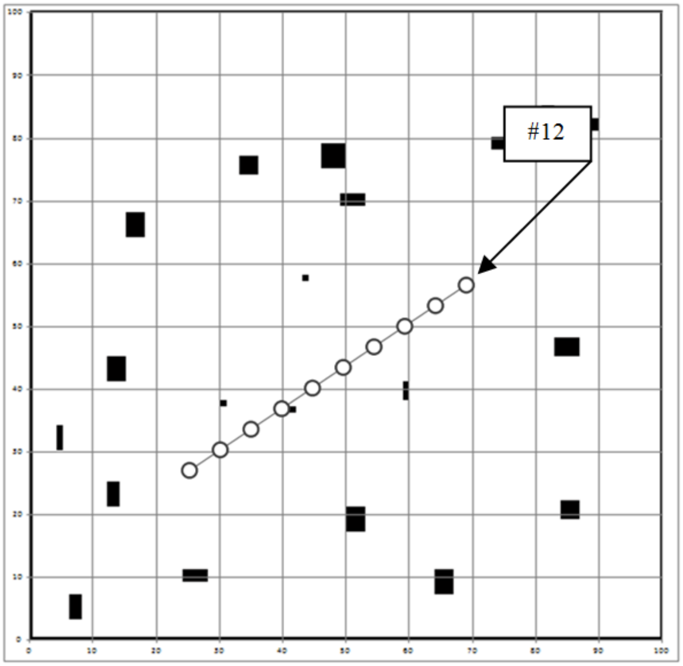


Figure 24: Smaller Collateral Objects - #12

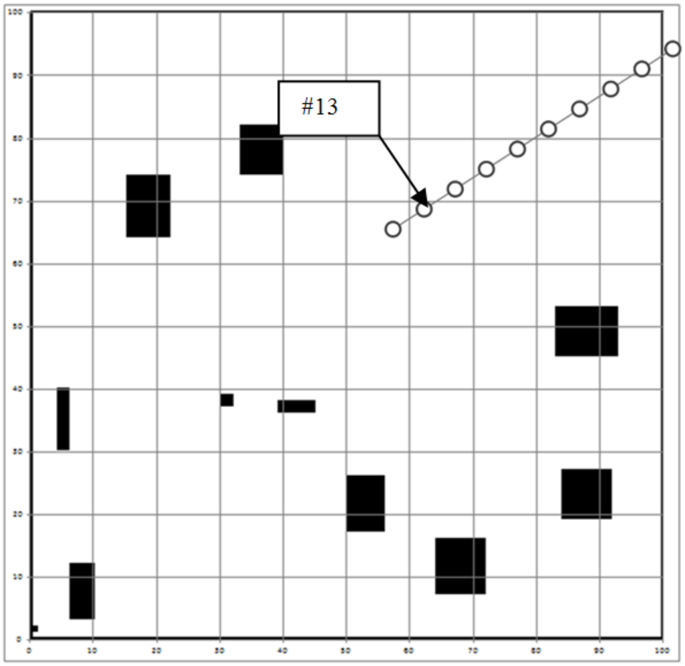


Figure 25: Fewer Collateral Objects - #13

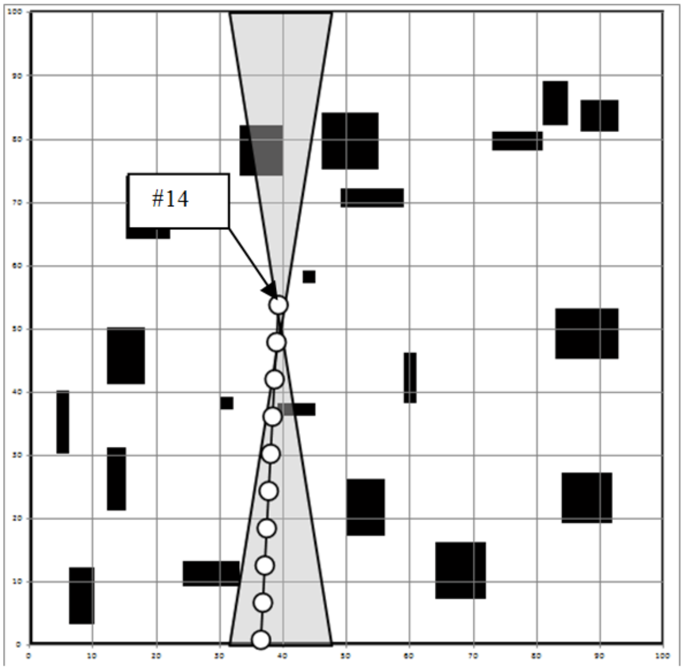


Figure 26: Constrained Flying Angle - #14

- While any change in the inputs may result in a vastly different area of the grid in which to drop, in the examples chosen we see that fewer bundles (#2), smaller bundle separation (#3), or smaller collateral objects (#12) did not greatly move the aimpoint from the standard case (#1).

- Changing the standard deviations (#4, 5, 6) not only has a tremendous effect on expected collateral damage, but can drastically change the location of the optimal solution.

- Changing the values of the collateral objects (#11) has a major effect on optimal location. This is true even if the average collateral object value is the same. Both the standard scenario (#1) and the varying value scenario (#11) have average collateral object value of 1, yet #11 has a 34% lower objective function value.

- Decreasing the size of the collateral objects (#12) yields little movement in the optimal aiming location from the standard case (#1), but over a 90% decrease in the expected amount of collateral damage.
- Fewer collateral objects (#13) results in 50% less of the scene covered by collateral objects and smaller collateral objects (#12) resulted in 89% less coverage of the scene. However, fewer collateral objects results in less damage indicating that it is better to have more collateral object area that is concentrated rather than less collateral area that is spread out.
- The constrained angle scenario (#14) demonstrates the potential risk when airdrop flight paths are restricted by weather, terrain, or other safety/logistics concerns. In this scenario, the flight paths are constrained to be within ten degrees of due north. From the standard scenario, we see a 29% increase in the expected amount of collateral damage.

2.5 Results and Discussion

In this paper, we present a characterization of the distribution of supply airdrops and methods for optimally dropping them. Specifically, supply airdrops follow a bivariate normal distribution in which the x and y deviations are uncorrelated ($\rho = 0$). A surrogate approximation function for the bivariate normal distribution supports quick integration of the distribution to assess drop risk. RSM with surrogate, and DE, both return Pareto optimum results depending on a tradeoff between runtime and accuracy. Both find near-optimal solutions of the non-linear program resulting from the airdrop problem, quickly finding settings for both an airdrop location and an approach angle. Enumeration is strongly dominated by all other algorithms.

It is natural to ask whether an expert eye is a substitute for algorithms. It turns out not. Suppose an airdrop planner who has been shown the oval shapes and scales of a bundled set of a supply airdrop could predict the optimal aimpoint within

50 meters in each direction and the drop angle within 5 degrees angle of the optimal solution. Note that this is a high standard - we have looked at hundreds of combinations of drops, yet still only approach that level of accuracy. In our base problem, where the collateral objects have the same weighting, the planner “eyeballing” a solution would have a collateral risk 14% higher than the optimum. In more complicated scenarios where the collateral objects are weighted differently, “eyeballing” a solution becomes much worse than the solutions found by our algorithms, with “eyeballed” solutions routinely worse by 20% or more. A more reliable technique must be implemented to limit damage and ensure recoverability.

In terms of future work, we have two ideas. First, rather than using bounds on the x and y directions to limit the bundle drop zone, an attractor function could be incorporated, which would approximate the likelihood of recovery as a distance from a given point (typically the middle of the scene). The attractor function would be weighted and added to the objective function for the problem. Second, automating chute selection when not all missions would be able to use the most accurate types of chutes has appeal. For example, in the case where an inventory of flights/chutes is available to cover a set of drop zones, we would optimize not just the individual drop but the portfolio of drops determining which chutes should be used for which mission to minimize the overall risk of collateral damage.

III. Pareto-Optimality for Lethality and Collateral Risk in the Airstrike Multi-Objective Problem

3.1 Introduction

Sources estimate that at least 6,000 to 9,000 civilian casualties [38] [98] [45] have occurred in Afghanistan since the beginning of the Global War on Terrorism (GWOT) as a direct result of Coalition military actions. More specific to the United States Air Force (USAF), over 1,000 civilian deaths have occurred since the inception of GWOT due to air strikes [38]. In addition, property damage resulting from airstrikes in Afghanistan to civilian-owned buildings has alienated some local residents and ruined goodwill created between NATO and anti-Taliban citizens [89].

The Chairman of the Joint Chiefs of Staff Instruction on No-Strike and the Collateral Damage Estimation (CDE) Methodology from 2009 [25] gives the U.S. Military its guidance on the subject of collateral damage. The document lays out such things as which types of buildings/structures are typically parts of a no-strike list and under which circumstances a commander may fire upon buildings known to contain collateral objects or people. The document touches on the use of human shields by the adversary, special restrictions on targets which may cause grave environmental or biological concerns, and the roles of personnel within the targeting process.

Of primary importance, the Instruction provides the collateral damage methodology (CDM) process which seeks to be “simple and repeatable” in order to provide “a reasonable determination of collateral damage inherent in weapons employment.” CDM is presented in five levels of increasing risk of collateral damage. A target will progress from level 1 until such point as it is no longer necessary to progress, thus making the target (and associated collateral risk) categorized as either CDE Level 1-Low, 2-Low, 3-Low, 4-Low, 5-Low or 5-High. Within CDM, weapons (and their method of employment) have assigned circular errors probable (CER) which give "a radius representing the largest collateral hazard distance for a given warhead, weapon, or weapon class considering predetermined, acceptable collateral damage thresholds that are established for each CDE level." From the CER, a collateral hazard area (CHA) is typically created by rotating this radius around the aimpoint of a weapon to create a circle. Collateral objects falling within that CHA for a given CDE level cause the target to be elevated into the next higher CDE level for further evaluation, until finally a CHA is created with no collateral objects within its boundaries, or the rating of CDE level 5-High is given to the target.

The CERs for given weapons and methods of employment only spell out the radius outside of which a collateral object should be safe from the weapon's firepower. This approach to the estimation of a weapon's power is known as the "cookie-cutter" approach, whereby all objects falling within the radius are considered to be destroyed and all targets falling outside of the radius are 100% safe. While simple and easy to implement, this assumption can be detrimental in the planning process for a weapons strike. The distribution, and not just the lethal range, of the weapon has a significant effect on the choice of an optimal aiming location and weapon selection. This paper seeks to quantify the effects of different weapon damage functions along with the effect of improvements made to the current policy guiding collateral damage mitigation.

3.2 Background

3.2.1 Collateral Damage Estimation.

In the literature the probability of destroying a point target is calculated with the following formula [64] [82]:

$$P = \int_x \int_y p(x, y) \cdot d(x, y) dy dx \tag{10}$$

where
P - probability that the point target is destroyed,
$p(x, y)$ - probability density function of the weapon's impact point,

$d(x, y)$ - probability that the point target is destroyed given that the weapon impacts at point (x, y).

In Equation 10 we indicate that lethality at a particular point (x, y) in space is both a factor of the uncertainty of the landing location $(p(x, y))$ and the damage caused at a given point in the case that it lands at a particular point in space $(d(x, y))$. Therefore, collateral damage estimations can be made by knowing the location error function and the damage function for the particular weapon.

The most commonly used location error formula for an air-to-ground weapon is [64]:

$$p(x, y) = \frac{1}{2\pi\sigma^2} e^{-(1/2\sigma^2)((x-\mu_x)^2+(y-\mu_y)^2)} \tag{11}$$

where

μ_x - x-coordinate of aimpoint,

μ_y - y-coordinate of aimpoint,

σ - standard deviation of the miss distance for the weapon.

This formula is the bivariate normal distribution, where the x and y miss distances are both uncorrelated $(\rho = 0)$ and the distributions in both the x and y directions are identical (that is $\sigma = \sigma_x = \sigma_y$) [29]. To account for situations where the miss distances in the x and y directions are different $(\sigma_x \neq \sigma_y)$, yet uncorrelated, one can use [34]:

$$p(x, y) = \frac{1}{2\pi\sigma^2} e^{-[\frac{(x-\mu_x)^2}{2\sigma_x} + \frac{(y-\mu_y)^2}{2\sigma_y}]} \tag{12}$$

For uncorrelated miss distances in the x and y directions $(\rho \neq 0)$ we must use a more complicated formula [32]:

$$p(x, y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{-[\frac{(x-\mu_x)^2}{2\sigma_x} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \frac{(y-\mu_y)^2}{2\sigma_y}]/2(1-\rho)^2} \tag{13}$$

A variety of damage functions are used for a variety of reasons, such as simplicity, accuracy in modeling the data, and ease of computation. Additionally, different types of air-to-ground weapons will have differing patterns of damage. The uniting characteristics of these functions is that they are decreasing functions as the radius (distance from point of impact) increases, their integral $\int_0^\infty d(r)dr$ is bounded, and they are "well-behaved" [64], meaning that either there exists a radius R such that for all $r > R$, $d(r) = 0$, or their function is continuous and monotonic.

$$\begin{aligned}
\text{Cookie-Cutter:} \quad d_1(r) &= \begin{cases} 1, & r \leq LR \\ 0, & r > LR \end{cases} && (14) \\
\text{Gaussian:} \quad d_2(r) &= e^{-r^2/2b_1^2} && (15) \\
\text{Exponential:} \quad d_3(r) &= e^{r/b_2} && (16) \\
\text{Lognormal:} \quad d_4(r) &= 0.5\{1 - erf[\frac{\ln(r/\alpha)}{\sqrt{2}\beta}]\} && (17)
\end{aligned}$$

The continuous damage functions come from 1-CDF of the probability functions (e.g. the exponential distribution which has a PDF of $f(r, b_2) = \begin{cases} b_2 e^{-b_2 r}, & r \geq 0 \\ 0, & r < 0 \end{cases}$, yields a CDF of $F(r, b_2) = \begin{cases} 1 - e^{-b_2 r}, & r \geq 0 \\ 0, & r < 0 \end{cases}$, which in turn creates the damage function $d_3(r) = 1 - (1 - e^{-r/b_2}) = e^{-r/b_2}$.) Typically, the lethal range of a weapon is calculated as $\int_0^\infty d(r)dr$ [64], thus in order to get a realistic comparison between damage functions, we must ensure that the lethal ranges using each of the damage functions is the same, therefore making it necessary to tweak the constants in the functions. For the exponential damage function b_2 is exactly equal to the lethal range of the weapon since $\int_0^\infty d_3(r)dr = b_2$. In the Gaussian damage formula the value for b_1 can be shown to equal $LR \times \sqrt{2/\pi}$. As noted in [82], since the lognormal damage formula has two inputs (α, β) there is no unique setting for the inputs to give a certain lethal range. For example, in Figure 27, the settings for the lognormal damage function are $\alpha = 0.615$ and $\beta = 1$, which can be found using the graph in Figure 27 for $LR = 1$. The PDF for each of the damage functions with a lethal range of 1 are depicted in Figure 28.

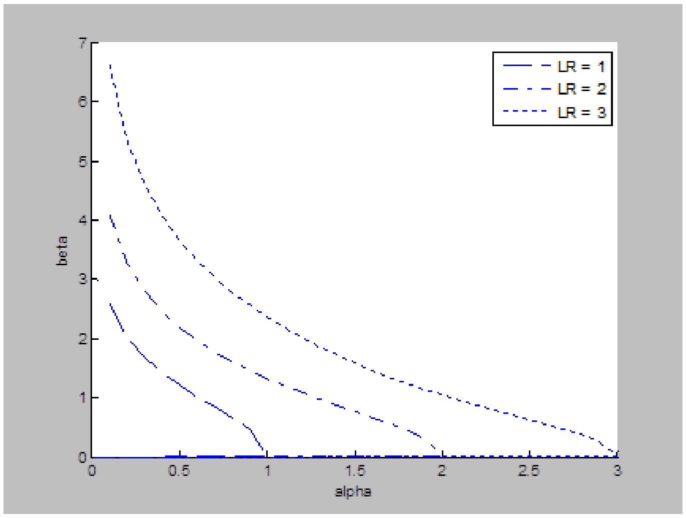


Figure 27: Lognormal Damage Function Inputs for Desired Lethal Range

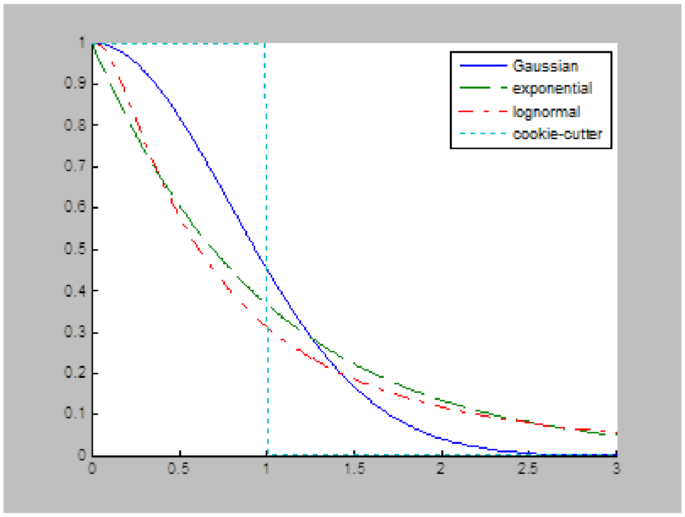


Figure 28: Damage Functions (LR=1)

3.2.2 Lethality Functions. Combining the damage function with the location error function (with the assumption of $\rho = 0$ and $\sigma_x = \sigma_y$) yields the following lethality function which is the expected amount of damage at a given distance from the aim-point (x', y'):

$$\int_X \int_Y p(x, y, x', y') d(x, y) dy dx \tag{18}$$

By converting (x, y) into a distance r from (x', y'), we can then inspect the shape of the lethality functions based on the different damage functions $d(x, y)$. In the case where the standard deviations are 1 unit and the lethal range of the weapon is 1 and 5 units, respectively, we generate the following graphs of the lethality functions:

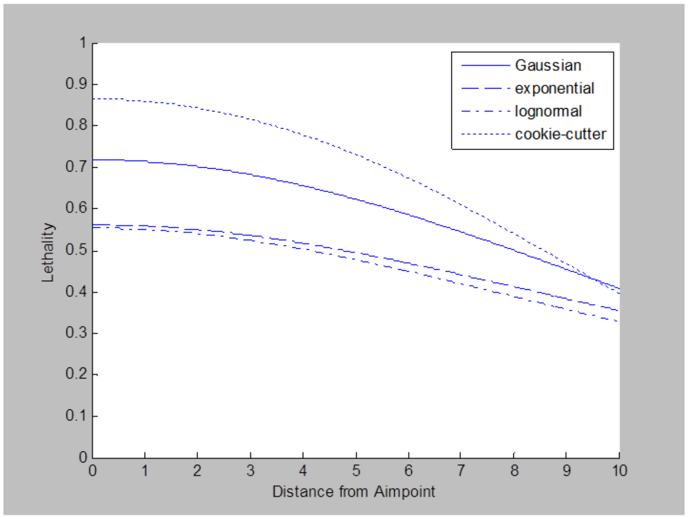


Figure 29: Lethality Functions (LR=1)

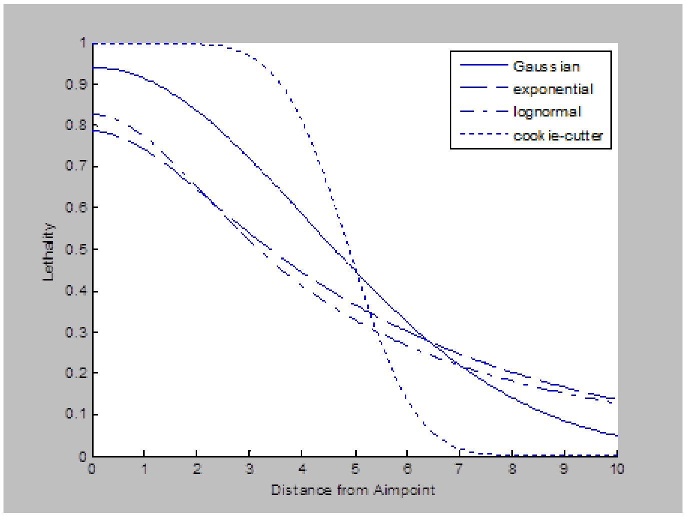


Figure 30: Lethality Functions (LR=5)

Figures 29 and 30 give insight as to which functions over- or under-estimate lethality and at which ranges. For example, in both graphs the cookie-cutter approach gives a higher result at the aimpoint ($r = 0$) than the other approaches. This is particularly pronounced in Figure 30 where the cookie-cutter approach yields lethality almost 75% higher than the lognormal and exponential damage functions at a distance from the aimpoint of 3. This phenomenon will always be most pronounced when the lethal range is high relative to the standard deviation of the miss distance of the weapon (in these two examples, the standard deviation of the miss distance for the weapon is 1). In the extreme case where the accuracy is degraded (yielding a high standard deviation) then the cookie-cutter approach will yield a smaller value at the aimpoint than the other approaches.

Lucas [64] goes into detail about the limiting behavior of each of the damage functions, noting that the lethality of the cookie-cutter function drops off the fastest when at higher distances from the aimpoint. This fact could be surmised from the very quick drop of the cookie-cutter approach in Figure 29 and particularly in Figure 30. Further, Lucas [64] proves that the lognormal damage function has higher lethality in its tail as r goes to infinity than any of the other damage functions, followed by the exponential, then the Gaussian, and finally the cookie-cutter approach, which has the lowest lethality in its tail. These insights about the limiting behavior of the lethality functions are irrespective of the accuracy of the weapon.

3.2.3 Offset Aiming. The concept of offset aiming is integral to the discussion of the minimization of collateral damage. Offset aiming is the concept that directly targeting military objects is not always optimal. For example, if a military target is located directly west of a collateral object (say, a school) and a given weapon striking the military target directly with the chosen weapon would carry enough force to significantly damage the school, then perhaps aiming slightly to the west of the military object would be optimal. The weapon and its lethality function might indicate that aiming 10 meters to the west of the military object would still employ enough firepower to accomplish the military objectives while the extra ten meters would put the school outside of the lethal range of the weapon, thus lowering any negative effects on the school.

Offset aiming is already part of the Department of Defense's (DoD) official policy on collateral damage estimation and mitigation. However, offset aiming is not considered until later levels of the CDE guidance. There is an argument to be made that offset aiming should be considered at all levels of the CDE process.

3.2.4 Weapons Employment as a Multi-Objective Problem. The current DoD policy on collateral damage indicates that collateral damage estimation must be performed prior to any pre-planned air-to-ground strike. A commander must be made aware of collateral risk in the area surrounding the military target and be provided with detailed collateral damage estimation before giving orders to strike. Efforts must be made to avoid collateral damage at a high cost, and the commander then decides if the amount of collateral risk is worth the military value of striking that target. Within CDE, differing weapons, aimpoints and methods of employment are considered in an attempt to satisfy military objectives in the face of collateral risk. The two concepts, military objectives and collateral risk, are played off against each other in order to create a mission plan that the commander is willing to support. La Rock [57] discusses a multi-objective approach towards weapon implementation taking into consideration collateral effects.

Typically, as the mission plan seeks to lower the collateral risk, the lethality on the military target suffers. The converse is also true, as the lethality on the target is sought to be increased, the risk to collateral objects in the area also increases. The lethality functions mentioned in the previous section are the same for both collateral objects and military targets; however, the goal is to minimize the lethality on collateral objects and maximize the lethality on military targets.

3.3 Formulation

For a given damage function $d(x, y)$ and a known delivery error function $p(x, y)$, we may begin to characterize the multi-objective function we seek to optimize for a given scenario.

$$\begin{aligned} \text{Goal: Max} \quad & f_1(x, y) \text{ (lethality on the military target)} \\ \text{Min} \quad & f_2(x, y) \text{ (lethality on collateral objects)} \end{aligned} \tag{19}$$

where

$f_1(x, y) = \int_X \int_Y p(x, y, x', y')d(x, y)dydx$

$f_2(x, y) = \sum_{i=1}^{n} c_i \int_X \int_Y p(x, y, x_i, y_i)d(x, y)dydx$

$p(x, y, x_i, y_i) = \frac{1}{2\pi\sigma^2} e^{-(1/2\sigma^2)((x-x_i)^2+(y-y_i)^2)}$

(x', y') - location of the military target

$d(x, y)$ - damage function for weapon

n - number of collateral objects in the area of concern

(x_i, y_i) - location of the i^{th} collateral object

c_i - value of the i^{th} collateral object

σ - standard deviation of weapon miss distance

Damage function specific inputs

α, β - lognormal damage function

b_1 - exponential damage function

b_2 - Gaussian damage function

LR - cookie-cutter damage function

The values of the collateral objects are typically subjective based on the desire to avoid striking them. The higher the value the more concern the planners have for avoiding this structure/area. The inputs to the lethality function are the lethal range of the weapon, the damage function to be used (along with choices of either alpha or beta for the lognormal function), and the accuracy of the weapon (the standard deviation of the miss distance).

3.3.1 Goal Programming Formulation. Once offset aiming is introduced to achieve collateral damage mitigation, a goal-programming approach can then be employed to get an accurate comparison between the damage functions' effect on both lethality and collateral damage. For instance, we could stipulate that the lethality on the military target must be above a certain number, say 90%. If this were our assumption, then our secondary goal would be to then find the point that satisfies this requirement while trying to minimize the collateral damage. We will call this approach the lethality first approach, or in this case the 90% lethality first approach. Conversely, if we wanted to use a constraint of no more than 10% collateral risk, we would start by eliminating all aimpoints that don't first satisfy this constraint. From there, we would then search the space that maximizes lethality on the military target; this will be the collateral first approach.

If we can turn either of the two objective functions into a constraint, then this problem is simply a single objective non-linear objective problem with an additional constraint from the other objective. There will now be just a single solution (in our case, a single point (x, y) in the scenario) which optimizes our objective.

The new formulation for the collateral first approach becomes:

$$\begin{aligned} \text{Goal:} \quad & \text{Max } f_1(x, y) \\ \text{subject to:} \quad & f_2(x, y) \leq c \end{aligned} \tag{20}$$

For the lethality first approach the formulation becomes:

$$\begin{aligned} \text{Goal:} \quad & \text{Min } f_2(x, y) \\ \text{subject to:} \quad & f_1(x, y) \geq c \end{aligned} \tag{21}$$

with the same constraints and definitions from Equation 20. In cases where the miss distance standard deviations in the x and y directions are the same, then f_1 will have the same value for all points which are the same distance from the location of the military target. Therefore, $f_1(x, y)$ can be thought of as $f_1(r)$ where r is the distance from the point (x, y) to the location of the military target (x', y'). That is, f_1 is symmetric about the location (x', y').

Solution techniques for solving non-linear programs such as response surface methodology and evolutionary algorithms are logical candidates. To further characterize our objective function, we observe that it is continuous, since each of the lethality functions are continuous (even the cookie-cutter lethality function). Therefore, non-linear optimization techniques which rely upon a continuous function should be tried, while techniques which are more suitable for discontinuous functions are less logical (such as tabu search, branch-and-bound, etc.)

3.3.2 Weighted Sum Scalarization. In a similar vein, to convert multi-objective optimization problems into single objective optimization problems, weights can be given to the multiple objective functions. In this case, the sum of the weighted function values is calculated in an effort to minimize (or maximize) the total. In our problem, since we seek to maximize lethality (f_1) and minimize collateral risk (f_2), then opposite signs are given to the two functions:

$$\begin{aligned} \text{Goal:} \quad & \text{Max } w_1 f_1(x,y) - w_2 f_2(x,y) \\ \text{subject to:} \quad & w_1 + w_2 = 1 \end{aligned} \tag{22}$$

The weighted sum scalarization approach presents a decision analysis problem since now we must construct weights for the value of collateral objects relative to the value of increased lethality on the military target. Of important note is that any solution to the weighted sum scalarization approach or any solution to the goal-programming approach will be a point on the Pareto front for the problem.

3.3.3 Multi-Objective Formulation. If we choose not to employ either goal-programming or weighted sum scalarization as a technique to combine the two objective functions into a single objective function (or a single objective function with an added constraint), then we can use a multi-objective optimization approach. When there is more than one objective function, there are (often) infinitely many solutions that lie on the Pareto front for the particular problem. Assuming that all objective functions attempt to minimize the response, the Pareto front is the set of solutions (x', y') such that there exists no other solution (x, y) for which $f_k(x', y') \leq f_k(x, y)$ for all k from 1 to the number of objective functions and $f_k(x', y') < f_k(x, y)$ for at least one value of k. These points in the Pareto front are non-dominated by any other solution.

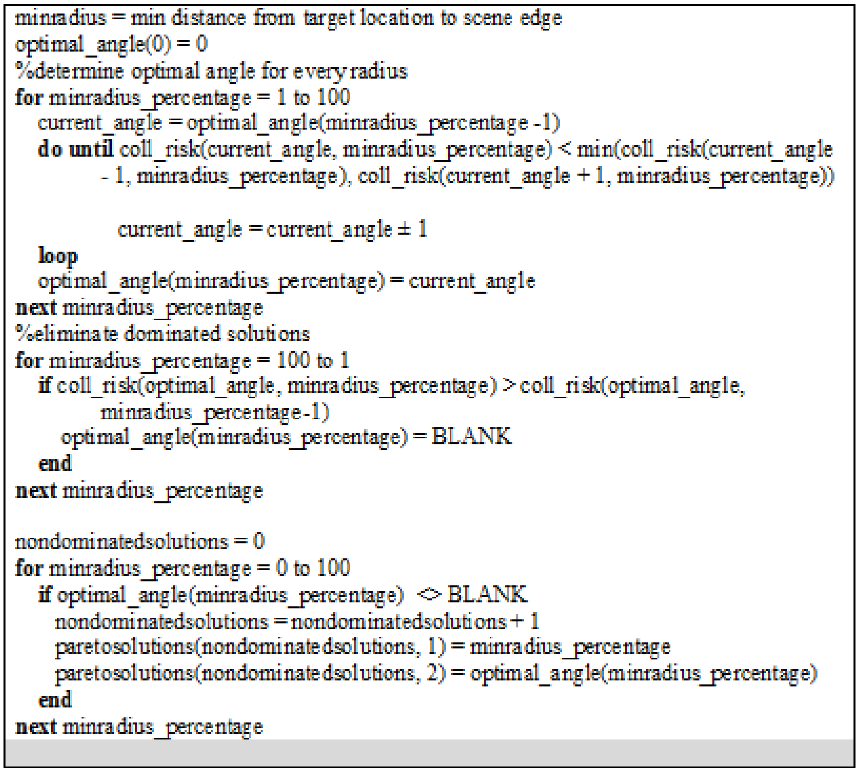
Algorithms to identify the entirety of the Pareto front are difficult to find, especially for problems where the objective functions are complex, such as in the collateral damage airstrike minimization problem. The majority of the literature on solving multi-objective optimization problems depends on evolutionary algorithms [19]. For instance, differential evolution approaches [2] [107] [104] [9] use mutation and combination methods developed for a single objective algorithm by Storn and Price [94]. We will compare the differential evolution algorithm of [2] against the algorithm we created to solve our multi-objective problem in the following section.

3.4 Methodology

In this section we will compare a differential evolution algorithm to a radius-based search method that leverages off the nature of the airstrike multi-objective formulation. The radius-based search method is shown to run in a fraction of the time of a differential evolution algorithm and produce better results. This radius-based search algorithm relies on the fact that with only a single military target in the region of interest, we can express the lethality function f_1 in terms of only the distance from the target. Thus, any point that is Pareto optimal must have the lowest collateral risk for all points the same distance from the target. The converse is not true; that is, a point which has the lowest collateral risk for all points the same distance from the target is not guaranteed to be Pareto optimal.

Further, while not guaranteed that all Pareto optimal solutions lay on a continuous line, in practice, we find that this is the case in nearly all scenarios. Since we are guaranteed the strictly decreasing nature of the lethality functions (regardless of their underlying damage functions), the location of the target is a Pareto optimal solution. Therefore, the continuous line emanates from the target location and extends to the edge of the scene. This is consistent with the graphs of the Pareto optimal solutions in the figures earlier in this section.

Figure 31 shows our radius-based solution algorithm which accurately estimates the Pareto front and Pareto optimal solutions.



```
minradius = min distance from target location to scene edge
optimal_angle(0) = 0
%determine optimal angle for every radius
for minradius_percentage = 1 to 100
   current_angle = optimal_angle(minradius_percentage -1)
   do until coll_risk(current_angle, minradius_percentage) < min(coll_risk(current_angle
           - 1, minradius_percentage), coll_risk(current_angle + 1, minradius_percentage))

             current_angle = current_angle ± 1
   loop
   optimal_angle(minradius_percentage) = current_angle
next minradius_percentage
%eliminate dominated solutions
for minradius_percentage = 100 to 1
   if coll_risk(optimal_angle, minradius_percentage) > coll_risk(optimal_angle,
           minradius_percentage-1)
      optimal_angle(minradius_percentage) = BLANK
   end
next minradius_percentage

nondominatedsolutions = 0
for minradius_percentage = 0 to 100
   if optimal_angle(minradius_percentage)  <> BLANK
     nondominatedsolutions = nondominatedsolutions + 1
     paretosolutions(nondominatedsolutions, 1) = minradius_percentage
     paretosolutions(nondominatedsolutions, 2) = optimal_angle(minradius_percentage)
   end
next minradius_percentage
```

Figure 31: Radius-Based Search Algorithm

3.4.1 Prototype Problem. Using the formulation laid out in the previous section, we may now begin to picture what the objective functions look like over the range of possible solutions. For the examples in this section, we will use the assumption that the scene is a 100 meter by 100 meter square. There is a single military target in each scene along with a number of equally weighted collateral objects that we seek to avoid damaging. With a miss distance standard deviation and a lethal range provided for the weapon used, contour plots can be developed for each of the damage functions. From these contour plots, it is very easy to identify where both the military targets and collateral objects are located within the scene (each of the contour plots is based on the same scenario). Figures 32 and 33 show the lethality functions and the collateral objects for our prototype problem.

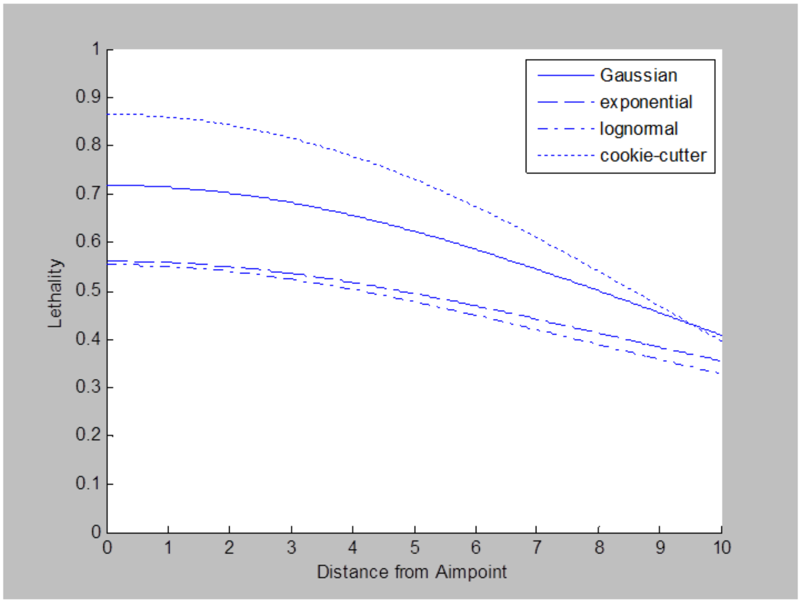


Figure 32: Lethality Function

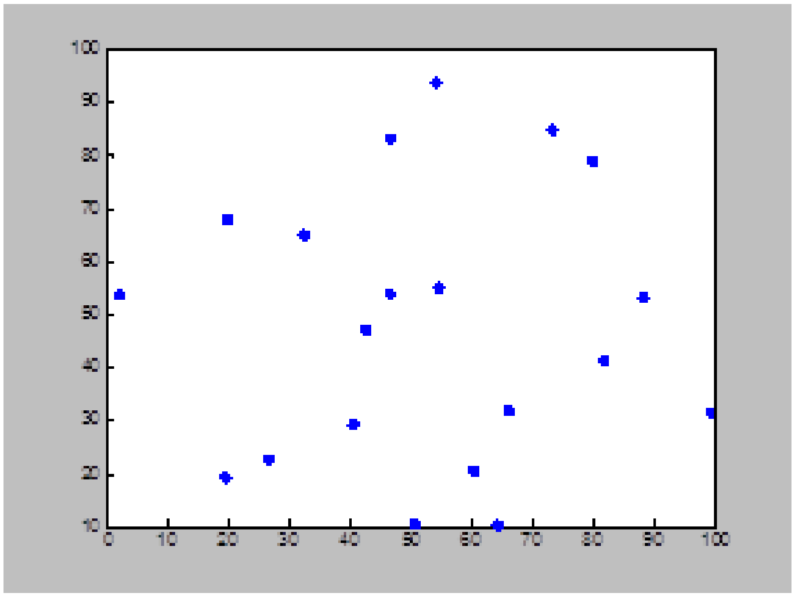


Figure 33: Location of Collateral Objects

To help visualize the nature of solutions, Figures 34 - 37 plot the two objective functions for the prototype problem using the four different damage functions (the lethality function is on the left and the collateral risk function is on the right). The cookie-cutter damage function (in Figure 36) shows the fastest drop-off of any of the damage functions yielding more distinct humps in the collateral risk graph surrounding the twenty collateral objects. Contrast this with the exponential damage function collateral risk graph (Figure 35) where the humps surrounding the collateral objects are much more blurred as a result of a more gradual decline in the lethality function for the exponential. The lognormal and Gaussian damage function graphs fall between the two extremes of the exponential and cookie-cutter graphs.

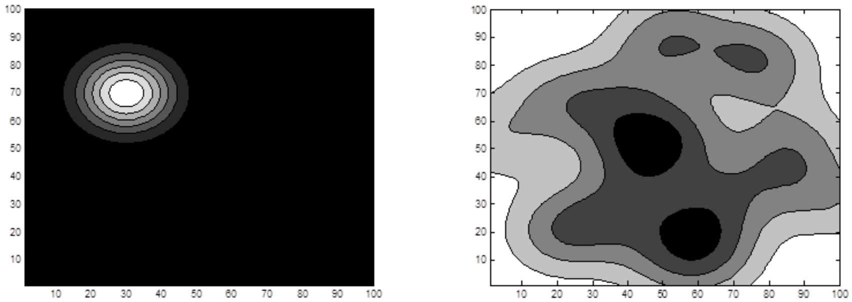


Figure 34: Gaussian Damage Function

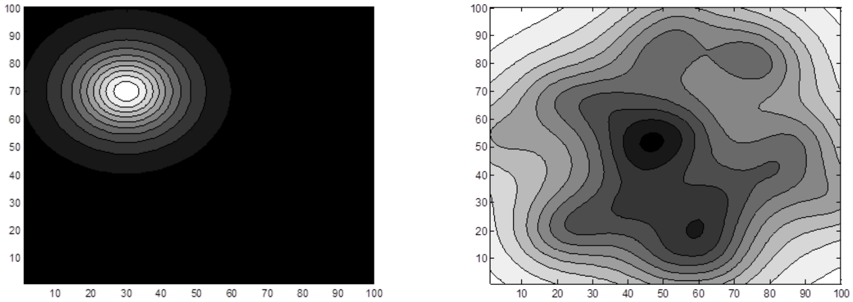


Figure 35: Exponential Damage Function

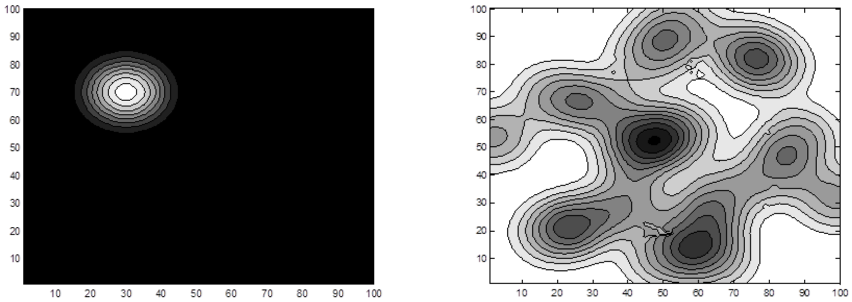


Figure 36: Cookie-Cutter Damage Function

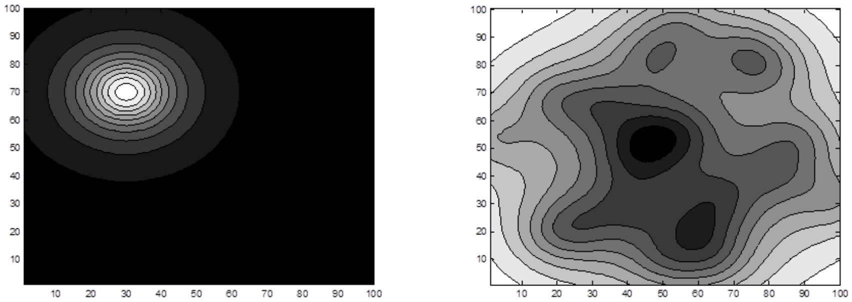


Figure 37: Lognormal Damage Function

Using enumeration (finding the lethality and collateral risk at a very fine resolution, 1000x1000, in the solution space), we find the non-dominated points. Figures 38 - 41 show the objective function values for all points in the scene (sampled every meter in the x- and y-directions) in the leftmost graphs with the higher lethality on the left edge and the lower collateral risk on the lower edge. The middle graphs show the non-dominated (Pareto-optimal) points in the objective space, which are those points on the "southwest" edge of the left plots. These points are non-dominated since no other point in this space has both a lethality value higher and a collateral value lower than these points. The right graphs show the location of the non-dominated points in the solution space.

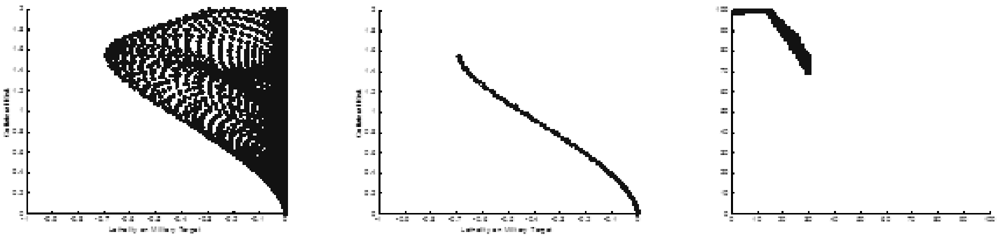


Figure 38: Gaussian Damage Function

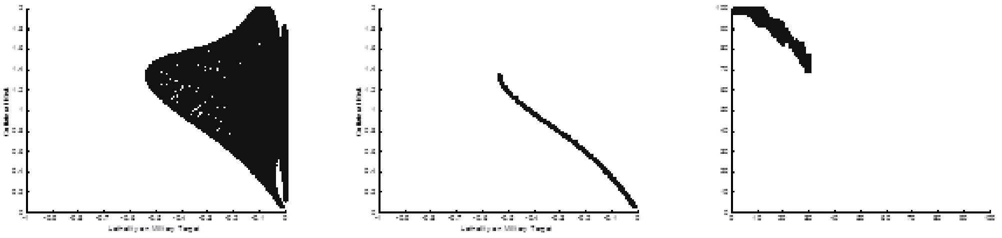


Figure 39: Exponential Damage Function

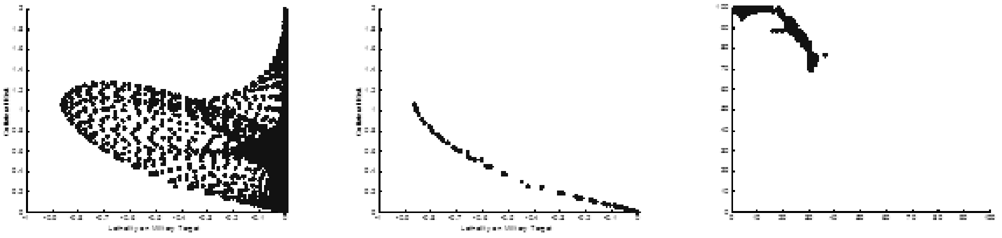


Figure 40: Cookie-Cutter Damage Function

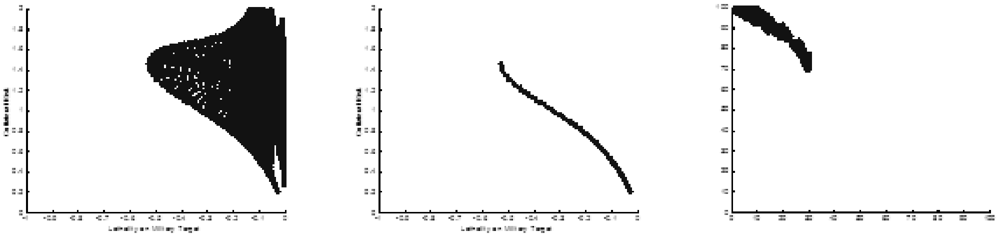


Figure 41: Lognormal Damage Function

One may not be surprised to see that the location of the military target is among the Pareto optimal points since no other point would have a higher lethality value than this point (due to the decreasing nature of the functions in Figure 29 and 30.) Keeping in mind that these functions were all tested with the same lethal range, accuracy and scenario, the cookie-cutter function estimates much higher lethality on the target and much lower collateral risk than the other damage functions. The Gaussian function (Figure 38) demonstrates significantly higher lethality on the target than the exponential and lognormal damage functions, but the collateral risk among these three damage functions is fairly comparable.

In the next section, we undertake the comparison of the algorithms discussed earlier and comparison metrics are introduced. The goal is to quickly and accurately locate the Pareto optimal solutions.

3.4.2 Algorithm Performance. Zitzler et al. [106] present methods for judging the effectiveness of algorithms for finding the Pareto front in multi-objective optimization problems. The first metric (accuracy) measures the minimum distance from the found solutions to a point on the true Pareto front (lower is better). The set of solutions found in the objective space are Y' and the true Pareto optimal frontier in the objective space is $\bar{Y}$:

$$M_1(Y') := \frac{1}{|Y'|} \sum_{a' \in Y'} \min\{\parallel a' - \bar{a} \parallel; \bar{a} \in \bar{Y}\}. \tag{23}$$

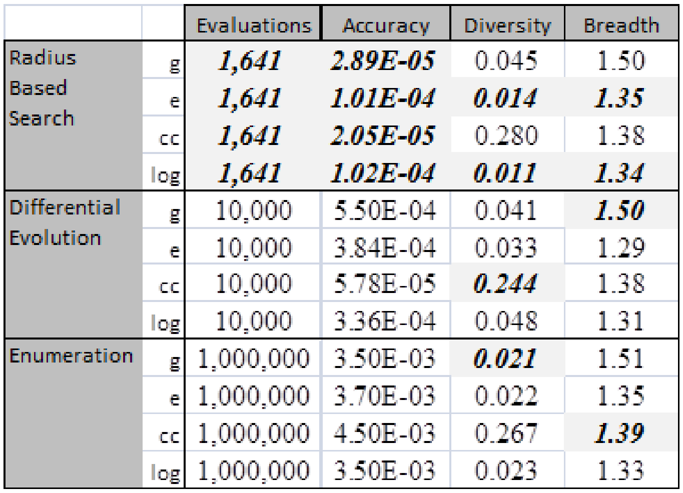
The second metric (diversity) measures the number of solutions found within a distance of σ from each found solution. This indicates how distinct found solutions are from one another (lower is better):

$$M_2(Y') := \frac{1}{|Y' - 1|} \sum_{a' \in Y'} |\{b' \in Y'; \parallel a' - b' \parallel < \sigma\}|. \tag{24}$$

The final metric (breadth) gives the maximum distance between found solutions for each coordinate. As noted in [18], for a two dimensional problem such as the collateral risk problem, this equals the distance of the two outer solutions (higher is better):

$$M_3(Y') := \sqrt{\sum_{i=1}^{m} \max\{\| a'_i - b'_i \|; a', b' \in Y'\}}. \tag{25}$$

Using these metrics, we compare the radius-based algorithm to enumeration and a differential evolution approach detailed in Figure 42.



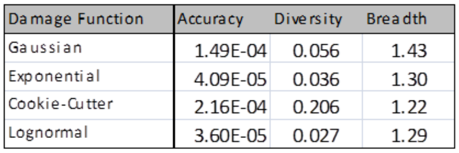
		Evaluations	Accuracy	Diversity	Breadth
Radius Based Search	g	***1,641***	***2.89E-05***	0.045	1.50
	e	***1,641***	***1.01E-04***	***0.014***	***1.35***
	cc	***1,641***	***2.05E-05***	0.280	1.38
	log	***1,641***	***1.02E-04***	***0.011***	***1.34***
Differential Evolution	g	10,000	5.50E-04	0.041	***1.50***
	e	10,000	3.84E-04	0.033	1.29
	cc	10,000	5.78E-05	***0.244***	1.38
	log	10,000	3.36E-04	0.048	1.31
Enumeration	g	1,000,000	3.50E-03	***0.021***	1.51
	e	1,000,000	3.70E-03	0.022	1.35
	cc	1,000,000	4.50E-03	0.267	***1.39***
	log	1,000,000	3.50E-03	0.023	1.33

Figure 42: Algorithm Performance (Prototype Problem)

Figure 42 compares the performance of our radius based search, a differential evolution algorithm, and enumeration (best results in bold). The radius based algorithm takes 37 minutes on a 2.60GHz machine to run, whereas the differential evolution approach takes almost four hours and the enumeration approach takes more than a week to compute. Our algorithm demonstrates the ability to generate points close to the true Pareto front for the prototype problem for all four damage functions as seen by the low values for metric 1. Metric 2, which measures the amount of points in the objective space within 0.001 of each other yields mixed results, with our algorithm generating the lowest totals for two of the four damage functions.

Further, small changes to the algorithm, such as using a logarithmic growth of the percentage value rather than linear growth, improve results for radius-based search for this metric. Metric 3 shows comparable results for the algorithms; the results are routinely within 5% of each other for a given damage function.

Figure 43 summarizes the metrics for 100 trials using our algorithm demonstrating performance of the algorithm against a variety of scenarios (accuracy ranging from 1 to 5 meters, lethal range varying from 5 to 20 meters, the number of collateral objects between 20 and 30, locations of the collateral objects and military targets varying within the 100m x 100m scene). These results confirm our algorithm's performance as seen in the prototype problem.



Damage Function	Accuracy	Diversity	Breadth
Gaussian	1.49E-04	0.056	1.43
Exponential	4.09E-05	0.036	1.30
Cookie-Cutter	2.16E-04	0.206	1.22
Lognormal	3.60E-05	0.027	1.29

Figure 43: Metrics across 100 Radius-Based Trials

Now that we are able to quickly and accurately find a broad range of Pareto optimal solutions for the true multi-objective formulation, finding solutions to both the goal programming and weighted sum scalarization formulation becomes straightforward. Recall that solutions to weighted sum and goal programming are a subset of the Pareto optimal solutions. Therefore, we must search only these solutions in order to find optimal solutions to the other formulations. The radius-based search gives us the location (x, y) and objective function values (f_1, f_2) for the optimal solutions; therefore, testing just these to find an optimal aimpoint is as simple as searching from among a small group of high quality solutions for the best values.

As long as the weights of the collateral objects are set to 1, the formulation becomes:

$$\text{Goal: Max } w_1 f_1(x, y) - w_2 f_2(x, y) \tag{26}$$

For example, using the Gaussian damage function, if we choose to value lethality (f_1) three times as much as collateral risk (f_2), then $w_1 = \frac{3}{4}$ and $w_2 = \frac{1}{4}$. Searching from our Pareto optimal solutions yields a best solution of 0.1658 ($f_1 = 0.69$, $f_2 = 1.41$) located at (29.4, 72.6). This solution is found without having to reevaluate the objective function. In a similar manner, assume we used a goal programming approach where we seek the lowest collateral risk while having at least 50% lethality on the military target for the same scene. We only need to find the Pareto optimal point that is the lowest lethality above 0.50, and that is our solution to this goal-programming problem. This point is located at (27.7, 77.6) with a collateral risk of 1.04 using the Gaussian damage function.

In the next section we will create Pareto front solutions for our airstrike problem using our radius based algorithm. In particular we explore optimal solutions based on different scenarios, guidelines and damage function in following sections.

3.5 Results

We first give a visual depiction of the effects of differing the damage function and approach on the location of the optimal solution. The scenario is the same as the one depicted in Section 3.4.1, with 20 collateral objects in a 100m x 100m scene. In the Figure 44 we zoom in on the area around the military target (located at (30, 70)). The lethal range of the weapon is 10 meters and the accuracy is 5 meters.

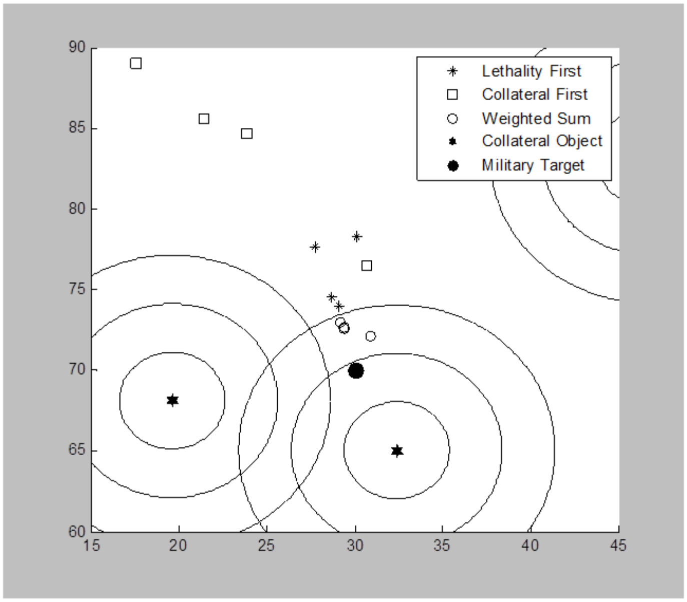
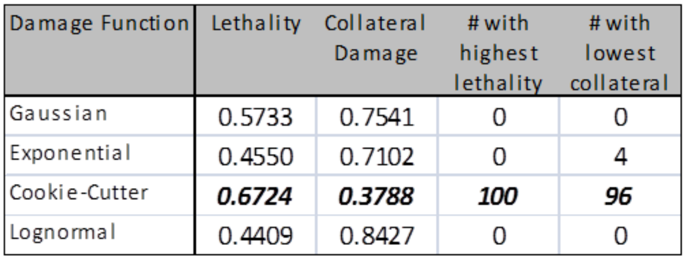


Figure 44: Location of Optimal Solutions

The effect of the two closest collateral objects (at points (32.4, 65.1) and (19.6, 68.1)) can be seen in Figure 44 as offsetting the optimal locations north of the military target for all guidelines and damage functions. As the collateral constraint increases it pushes the aimpoint away from the target and, similarly, as the lethality constraint increases, the closer the optimal aimpoint becomes to the target (in Figure 44, we show a lethality first constraint of 50% and a collateral first constraint of 50% for comparison).

3.5.1 Effects of Differing Damage Functions. With the differing shape due to the damage functions seen in Figure 28, the result using a zero offset distance with a lethal range between 5 and 10 meters along with a standard deviation of the miss distance of 5 meters and randomly generated collateral objects, we see the following results for 100 trials:



Damage Function	Lethality	Collateral Damage	# with highest lethality	# with lowest collateral
Gaussian	0.5733	0.7541	0	0
Exponential	0.4550	0.7102	0	4
Cookie-Cutter	***0.6724***	***0.3788***	***100***	***96***
Lognormal	0.4409	0.8427	0	0

Figure 45: Comparison of Damage Functions

The cookie-cutter damage function which is used in CDE overstates lethality by 37% compared to the average of the other three damage functions and it understates collateral risk by 51%. In the next section, we present a more general result.

3.5.2 Theoretical Collateral Function Values for Differing Damage Functions.

In a space where collateral objects are randomly located throughout an infinite space, we are able to calculate the theoretical collateral function values for the different damage functions if we know the number of objects per unit of planar space. By rotating the lethality function around the x-axis and multiplying by the number of targets per square meter (n), we would obtain expected collateral value for a randomly generated scene:

$$E[f_2(r)|n] = 2\pi n \int_0^\infty r d(r) dr \tag{27}$$

When looking at the theoretical value for each damage function, we can see how the damage functions will give wildly different collateral risk values. $E[f_{2_{cc}}] \leq E[f_{2_g}] \leq E[f_{2_e}]$ regardless of the accuracy and lethal range of the weapon in a randomly generated infinitely large scene.

That is, the expected collateral risk for the cookie-cutter damage function will always be less than the Gaussian damage function which will be less than using the exponential damage function (proof in Appendix A). The collateral risks for varying lethal ranges are shown in Figure 46, where the theorem is demonstrated for the varying damage functions.

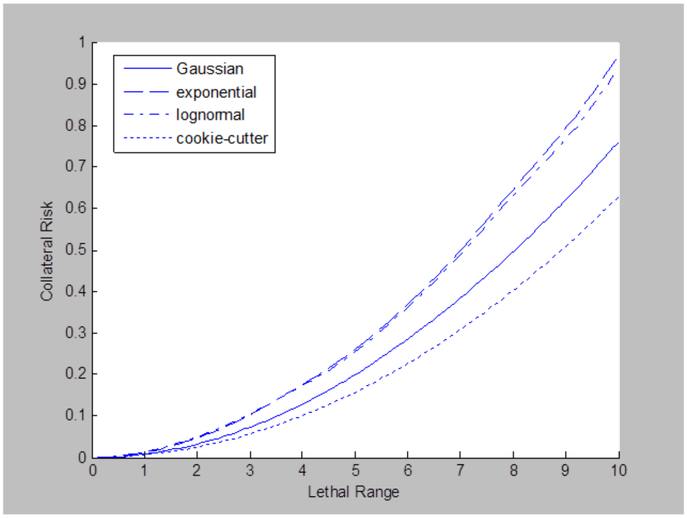
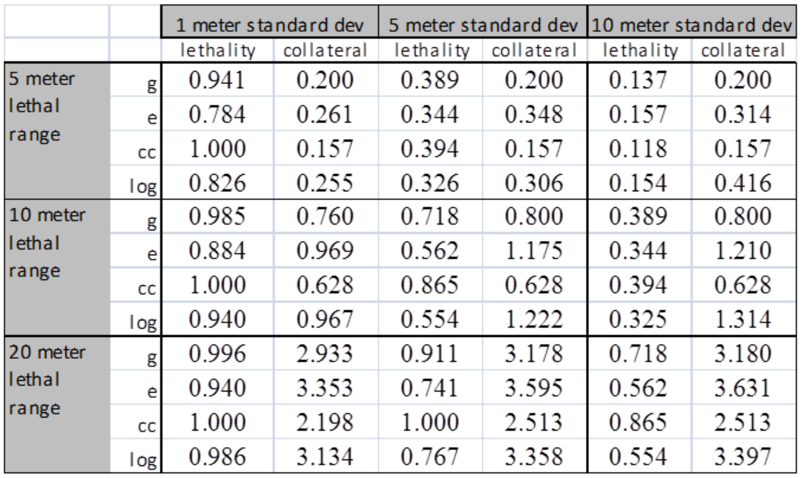


Figure 46: Collateral Risk by Damage Function

The values in Figure 47 summarize the collateral damage for twenty collateral items within a 100m x 100m square along with the lethality at the aimpoint for the various damage functions.

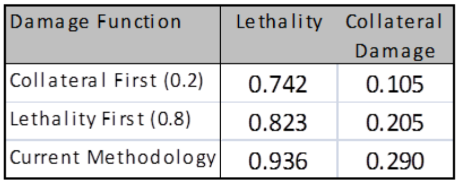


		1 meter standard dev		5 meter standard dev		10 meter standard dev	
		lethality	collateral	lethality	collateral	lethality	collateral
5 meter lethal range	g	0.941	0.200	0.389	0.200	0.137	0.200
	e	0.784	0.261	0.344	0.348	0.157	0.314
	cc	1.000	0.157	0.394	0.157	0.118	0.157
	log	0.826	0.255	0.326	0.306	0.154	0.416
10 meter lethal range	g	0.985	0.760	0.718	0.800	0.389	0.800
	e	0.884	0.969	0.562	1.175	0.344	1.210
	cc	1.000	0.628	0.865	0.628	0.394	0.628
	log	0.940	0.967	0.554	1.222	0.325	1.314
20 meter lethal range	g	0.996	2.933	0.911	3.178	0.718	3.180
	e	0.940	3.353	0.741	3.595	0.562	3.631
	cc	1.000	2.198	1.000	2.513	0.865	2.513
	log	0.986	3.134	0.767	3.358	0.554	3.397

Figure 47: Expected Collateral Damage

These results mirror the results of [64], which indicates that the lognormal damage function gives a higher collateral risk value (and thus a longer stand-off range) relative to the other damage functions and conversely the cookie-cutter function gives a lower collateral risk than the other damage functions and therefore a shorter stand-off range.

3.5.3 Effects of Differing Guidelines. In this section we test another 100 randomly-generated real-world representative problems. Current policy allows for no offset aiming, whereas the two other guidelines allow for offset aiming while using either collateral risk or lethality as a constraint, as seen in Equation 21 and 22. We test lethal range of 5 meters and accuracy of 1 meter using the cookie-cutter damage function. Further, we assume that the lethality must be at least 0.8 for lethality first and collateral risk must be no more than 0.2 for collateral first (results in Figure 48).



Damage Function	Lethality	Collateral Damage
Collateral First (0.2)	0.742	0.105
Lethality First (0.8)	0.823	0.205
Current Methodology	0.936	0.290

Figure 48: Results for $\sigma = 1$, $LR = 1$

The baseline methodology, which aims directly at the target, yields the highest lethality. However, the collateral damage is, on average, 176% higher than for the collateral first approach and 41% higher for the lethality first approach, with the lethality being 26% and 14% higher versus the collateral first and lethality first approaches, respectively.

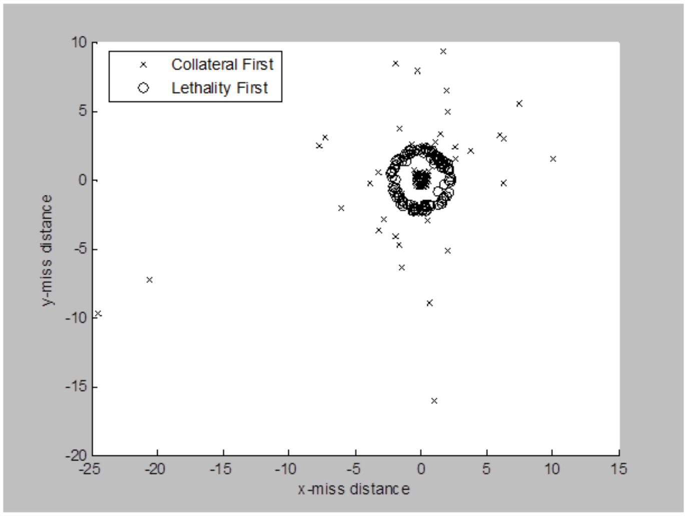


Figure 49: Location of Solutions

Figure 49 shows the location of solutions. The lethality first approach requires an optimal location which is within four meters of the target location in order to have at least 80% lethality on the target. This is not the case for the collateral first approach, which can result in solutions very far from the military target in order to find a location that falls below the 20% threshold for collateral risk (as seen in Figure 49 where one of the optimal aimpoints is located at roughly (-25, -10) a distance of almost 27 meters from the target).

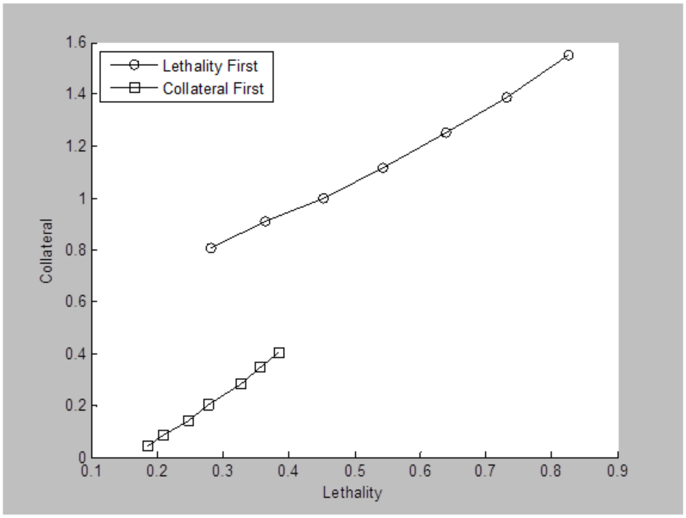
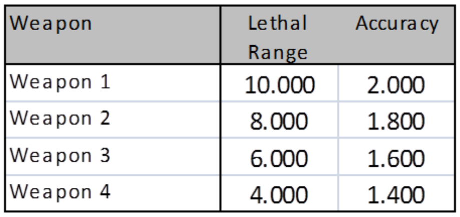


Figure 50: Lethality First vs. Collateral First

In Figure 50, we compare the lethality first and collateral first approaches (using the Gaussian damage function) for varying levels, with the collateral constraint and lethality constraint ranging from 0.2 to 0.8 and differing lethal ranges and standard deviations. In some scenarios, there are no locations which will yield a collateral risk less than the constraint or lethality above the lethality constraint. Thus, feasibility of solutions satisfying both goals is not assured. The trade-off also introduces the idea of ordnance selection, the topic of the next section.

3.6 Ordnance Selection

Airstrike planners may have a variety of weapons as well as methods of employment (fusing, run-in, etc.) which affect the lethal range and accuracy. Thus, to increase lethality and decrease collateral risk a planner must not only look for the optimal aim-point but also the best selection of weapon and employment. For instance, assume that in the same scenario in Figure 33, the planner was presented with the following options of weapon and employment:



Weapon	Lethal Range	Accuracy
Weapon 1	10.000	2.000
Weapon 2	8.000	1.800
Weapon 3	6.000	1.600
Weapon 4	4.000	1.400

Figure 51: Weapon/Employment Parameters

Let us assume that the damage function follows an exponential distribution. The lethality/collateral risk trade-off values appear in Figure 52. From this figure, we see the smaller, more accurate weapons yield a slightly lower lethality, but with a large reduction in collateral risk (a 75% reduction in collateral risk with only a 10% reduction in lethality when moving from Weapon 1 to Weapon 4). While these numbers will vary with the scenario and damage function, the take-away is that smaller weapons provide nearly as much lethality as larger weapons, but with a greatly reduced amount of collateral risk as long as the accuracy also improves with the smaller weapon.

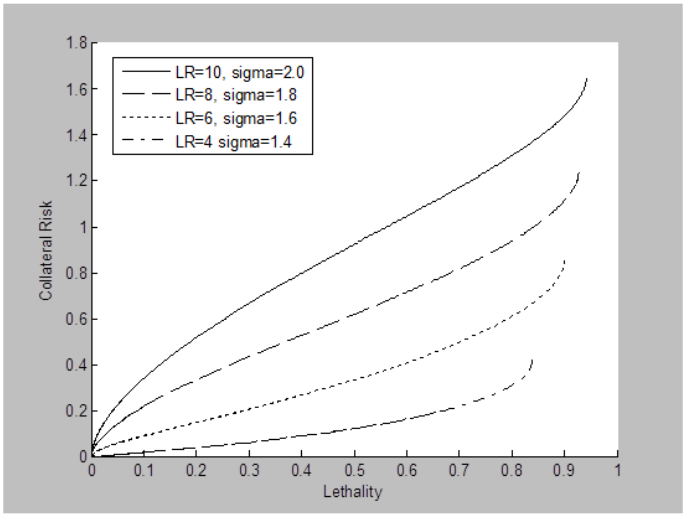


Figure 52: Weapon Lethality and Collateral Risk

Moreover, assume that the mission planners would like to have 90% lethality on a military target while minimizing collateral risk. If they had only weapons 1 and 4, only weapon 4 aimed close to the target would reach this goal and that would be at a significant cost (collateral risk around 1.5). However, if they could fire multiple weapons, perhaps they would choose firing two weapon 4's, offsetting the weapons to achieve a lethality just over 0.7 each (since $0.3^2 \approx 0.1$ yielding a 10% likelihood of not destroying the target). This choice would yield a combined collateral risk of under 0.4, over a 70% decrease in collateral risk from firing weapon 1.

Since planners will often be confronted with relatively few choices in terms of weapons and employment options, then the evaluation of the collateral risk, lethality and optimal aimpoint for each can be performed in an enumerative manner (this can be accomplished in parallel if time is a concern). The search algorithms can be used when either a clear goal is stated (e.g . minimize collateral risk while ensuring 90% lethality on the target) or when a variety of alternatives is available to the decision-maker. It is, in fact, a straight-forward task to include weapon selection in our formulation if desired.

3.7 Summary and Conclusions

In this paper we develop a quick and accurate algorithm for accurately creating the Pareto optimal frontier in the multi-objective airstrike problem. This algorithm, which leverages specific attributes of lethality and collateral risk, is shown to routinely outperform differential evolution and enumeration algorithms. Once Pareto optimal solutions are found these can be quickly converted to solutions to the associated goal-programming or weighted sum scalarization problems. The choice of damage function is shown to greatly affect the expected lethality and collateral risk in an airstrike underscoring the need for accurate estimation of weapons effects.

We demonstrate that the current methodology of not using offset aiming yields lethality 26% higher at a cost of collateral risk 176% higher than a collateral first approach. The algorithm presented can be incorporated into the weapon (and employment) decisions facing an airstrike planner, who could alter selections based on the minimum lethality needed or maximum collateral risk allowed to remedy the limitation from non-offset targeting. Future work will see the application of our algorithm to more complicated lethality and collateral risk models such as the JWS and JMEM tools currently used by the USAF.

IV. Look-Look-Shoot: Finite and Infinite Horizon Markov Decision Policies with Limited Intelligence

4.1 Introduction

In fast moving troops-in-contact (TIC) situations, information is often subject to the fog and friction of war. Forces cannot wait for perfect information and, as a result, mistakes are made, civilians and even friendly forces are killed. However, the alternative of waiting for "perfect" information before an airstrike is ordered has its own set of consequences. Ground forces may be pinned down, and every second that goes by increases their likelihood of being shot or killed by enemy forces. However, there is a cost to making the wrong decision which could result in civilian casualties, friendly force casualties and unnecessary collateral damage. The questions then become, "How long do we wait for perfect information?" or "When have we received enough imperfect information to make a decision?"

In these situations, there will always be a trade-off between the cost of civilian casualties and the cost of losing friendly forces. There will be a cost of abandoning friendly forces when they need help, there will be an opportunity cost of air support spending time in an engagement that does not truly threaten them, and there will be a cost of killing non-combatants.

4.1.1 Data Fusion. A goal of all observers in a conflict is to correctly classify the nature of the suspected enemy in a timely manner (as time costs lives, money, and the opportunity to support other engagements). In pre-planned missions, where the nature and location of suspected enemy forces are well known, there have been relatively few civilian casualties in recent years (only two pre-planned missions resulted in civilian deaths from 2006 to 2007). Conversely, civilian casualties from TIC situations have exploded in recent years (over 400 civilian deaths from TICs in 2006-07). TIC situations are defined by the lack of previous intelligence about the suspected enemy, the location of friendly, neutral and enemy forces, the terrain, and the capabilities of friendly and enemy forces.

In a TIC situation there may be a variety of observers attempting to determine the true nature of the suspected hostile player. UAVs circling overhead sending images of a building back to image analysts, ground forces in varying locations relative to the suspected enemy, and air support pilots viewing the scene from above will all give a unique picture of the battlespace. Each will have their own unique assessment of the ground scenario, and we can assume that each of them has a differing likelihood of being correct. In order to synthesize these perspectives the field of data fusion must be introduced to the problem as we seek to get the most correct information out of the imperfect data gathered from these sources.

Polikar [81] discusses the idea of ensemble based systems in decision making, whereby diverse classifiers making individual classifications are fused together to develop a cleaner picture of an unknown event. Polikar gives as an example a patient undergoing tests for a neurological disorder, who might undergo an MRI scan, EEG, blood and other tests. An individual test alone might give a prediction as to whether the patient has the disorder or not, and each test has some type I and type II error involved with it. The reason multiple tests are performed is that as the doctors get varying pictures of the disorder, they will make a more robust classification yielding lower type I and type II errors. We seek to give a framework for applying these same ideas to TIC classification.

In the simplest case, the observers in a TIC are trying to determine if a building or group of individuals constitute a legitimate military target. The suspected enemy forces might truly have combatants among them, but often the multitude of civilians (non-combatants) among them who would be put in danger with an airstrike could outweigh the gain of killing the combatants. Recent military guidance has instructed US personnel to exit situations in which non-combatant lives are in danger, even if it means disengaging with known enemy forces, if the personnel are able to safely exit the environment [39].

4.1.2 Decision Making with Imperfect Information. All decision making takes places without accurate or complete information on the outcome of the decision. Often, with more time studying a decision, better information comes to light giving the decision maker a better grasp of the true nature of the problem and the effects of a decision [56] [85] [80]. While studying the time pressure on decision making, Payne [79] notes, "In some cases, the longer the delay in making the decision, the lower the expected return (value) from even the most accurate of decisions." The fact is that we cannot always afford to make an accurate decision, when doing so delays making a "satisfactory" decision.

Decisions are routinely made in a sequential manner. Consider the stock market, where once an investor purchases a stock, every following day, he may choose whether to sell that stock he purchased, buy more of that stock he purchased, or simply do nothing. Day after day, a decision is made, and the optimal decision process is one in which the profit made on the investment is maximized at some point in the future and the decision made on any given day depends on all of the decisions leading up to that point (e.g. the investor cannot sell stock on a day if he sold all of his stock the day prior).

The process of sequential decision making can be analyzed with dynamic programming [31]. We can see the applicability of dynamic programming to TIC situations, where after receiving incremental information, a pilot may choose to either fire upon a target, continue to loiter above a target or to leave the situation and attend to other potential targets. However, as in the stock example, the pilot's decision depends on the decisions he has previously made. A target cannot be struck if it has been previously struck (if we assume a strike completely destroys the target) and a pilot cannot fire upon a target if his previous decision was to leave the scene. Just as with the investor, the pilot's goal is to link together the chain of decisions which will yield the best outcome at some defined future point. In order to find that decision chain we must solve a dynamic program.

4.1.3 Dynamic Programming. Dynamic programming can be thought of using stages and states. A stage is a discrete point in time at which a decision (u_k) is made based on the state (x_k) of the system. The state is the summary of all decisions from previous stages and their outcomes, due not only to the decisions made but also the randomness (w_k) involved with moving from stage to stage. Some additive reward is gained for each decision made, and the goal is to maximize the sum of the rewards over the time horizon of the problem. Bertsekas [13] lays out the main ingredients of a basic dynamic programming formulation as:

1. A discrete-time system of the form $x_{k+1} = f_k(x_k, u_k, w_k)$,
2. Independent random parameters,
3. A control constraint (decision),
4. An additive cost of the form $Eg_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k)$,
5. Optimization over policies (rules for choosing u_k for each k and each possible value for x_k).

Denardo [31] formulates the dynamic programming problem as $x_{k+1} = f_k(x_k, u_k, w_k)$, $k = 0, 1, \ldots, N-1$ where:

k indexes discrete time,

x_k is the state of the system and summarizes past information that is relevant for future optimization,

u_k is the control or decision variable to be selected at time k,

w_k is a random parameter,

N is the number of time control is applied.

For the Look-Look-Shoot (LLS) problem, we assume that each stage brings about more data relating to the problem assuming the pilot continues to loiter. The rewards in this model are typically assumed to be costs, either the cost of a "look" decision or the likelihood of being incorrect given a "leave" or "shoot" decision.

4.1.4 Markov Decision Process. A Markov decision process (MDP) is a problem in which there is a decision maker, a finite number of policies or choices the decision maker can choose, a transition probability matrix which defines the likelihood of the next state given the current state and policy, a transition reward matrix which indicates the current reward gained for the state and policy, and a performance metric based on the rewards gained during the stages of the MDP [42].
S, a finite state space of possible system states. A realization of the random variable S is denoted by s.
A, a finite set of actions. A realization of the random variable A is denoted by a. An action a causes transitions from the current state to some new state.
$T : S \times A \times S \to \mathbb{R}_{[0,1]}$ is the state-transition function, giving the probability that the agent transit to state s' when it is in state s and takes action a. In other words, the transitions specify how each of the actions and exogenous events change the state of the world. We denote by $T(s, a, s') = P(s'|a, s)$ this probability. We have for each s, $\sum_{s'} P(s'|a, s) = 1$.
$R : S \times A \to \mathbb{R}$ is the reward function giving the expected immediate reward gained by the agent for taking each action in each state.

4.1.5 Partially Observable Markov Decision Process. When the agent is unsure of the state s that he is currently in, unlike the MDP where the agent knows where he is at all times, this problem becomes a POMDP. There is some probability distribution around the state in which the agent thinks he is in (the belief state b).

$O : S \times A \to \Pi(\Omega)$ is the observation function, which gives, for each action and resulting state, a probability distribution over possible observations (we write

$O(s', a, o)$ for the probability of making observation o given that the agent took action a and landed in state s' [72].)

Monahan [72] introduces a system where one of three decisions may be made. Either the observer can "inspect" - attempt to observe the true state of the target another time (at a cost), "stop" - make a determination as to the true state of the target and have no option for further observation, or "continue" in which he moves to the next time interval (at a cost) where the same three options will again be available to him. In the next time interval there is some probability that the nature of the target has changed which is a difference from the assumptions in this paper. McAllister [68] further notes, "While the Markovian property does not hold for the state of the system, it does hold for the belief state of the system. The optimal policy for any given stage is only dependent on the current belief state and not decisions made in previous stages."

4.1.5.1 Two-State Belief State.

When trying to classify a suspected target in a TIC situation, we are concerned with classifying the target as "legitimate" or "illegitimate", hence the two-state belief state (we believe, with some likelihood that the target is "legitimate" or "illegitimate"). In a POMDP with a two-state belief state, the likelihood of being in either of the two states can be expressed by the belief (p) of being in one state or the other, since the likelihood of being in the other state is $1 - p$. In Figure 53, the current belief state is expressed as p, which is the observer's belief that the true nature of the state is s_1. As the observer becomes more confident in s_1 being the true nature of the state, then p will increase, and, similarly, if the observer becomes more confident that s_2 is the true nature of the state then p will decrease (as s_2 becomes the more likely true state). Thus, our belief state can be written as $b = (p, 1 - p)$ yielding $b(s_1) = p$ and $b(s_2) = 1 - p = q$ (note: $b(s_1) + b(s_2) = 1$).

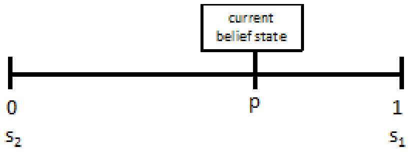


Figure 53: Two-State Belief State

In a simple example, depicted in Figure 54, initially the observer believes that the true state is equally likely to be in s_1 or s_2. The observer knows that each observation has a 75% likelihood of being correct, thus, if he observes s_1, then his belief state is $b = (0.75, 0.25)$ and if he observes s_2, then his belief state becomes $b = (0.25, 0.75)$.

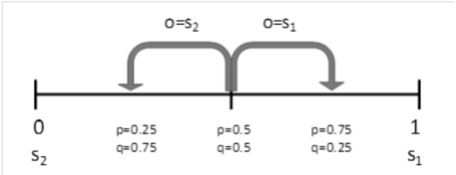


Figure 54: Belief State after First Observation

A complete policy for a POMDP is the optimal policy for each possible belief state [50]. The optimal policy for a given stage is then only dependent on the belief state at that stage and not decisions made during previous stages (the Markov condition). Of particular note, a belief space may be partitioned into more regions than actions, meaning that one action can be optimal for disparate regions of the belief space [50] [73] [72].

Working from Bayes' Theorem, the new belief state at a new stage is:

$$b'(s') = \frac{O(s', a, o) \sum_{s \in S} T(s, a, s') b(s)}{P(o|a, b)} \tag{28}$$

with $P(o|a,b) = \sum_{s \in S}[O(s',a,o)\sum_{s \in S} T(s,a,s')b(s)]$ indicating that future belief state is a function of both the current stage's belief state, the action taken during the current stage, and the observation made.

4.1.6 Shoot-Look-Shoot. Glazebrook and Washburn [41] present a review of the "Shoot-Look-Shoot" problem in which a marksman is required to kill a given number of targets. Once the marksman shoots, he looks to see if the target has been killed, and then, if the target hasn't been killed, he may choose to shoot again at that same target. The problem gets further complicated by imperfect information wherein the marksman may get possibly incorrect information as to the alive/dead status of the previous target. Glazebrook and Washburn view the problem as a Markov decision process and use a stochastic dynamic programming approach to develop the marksman's optimal strategy.

A difference between the "Look-Look-Shoot" problem and the "Shoot-Look-Shoot" problem is that LLS allows for only one shooting. We assume that when a target is aimed at, it is completely destroyed; battle damage assessment is not implemented. In the LLS problem, imperfect information plays a role when the pilot is unsure whether a target is a legitimate military target or not.

4.2 Finite Horizon

When viewing the LLS problem as having a finite horizon we allow for only a given number of stages, at which time the TIC situation has been resolved, either by firing upon the target or the air support leaving the situation. The finite horizon problem lends itself to being solved through the dynamic programming method of recursive fixing starting from the final time period and incrementally making decision backwards. We implement this method in both cases where the quality (likelihood of being correct) of information is constant across stages and where information improves as we move through the stages as more surveillance and intelligence become available in the TIC situation.

We will further assume that information arrives at set intervals each one unit of time apart. The inputs to the problem then become: the information distribution as a product of time, the cost of waiting one cycle for more information (c_w), and the cost of striking a building which is not a legitimate military target (c_s). Further $c_w < c_s$ since otherwise there would never be an incentive to wait for more information also the choices for action at each stage is either "L" (look, make another observation), "S" (shoot, fire upon the target ending the scenario) or "X" (exit, leave the scenario without firing). Note: only the "L" choice results in future stages.

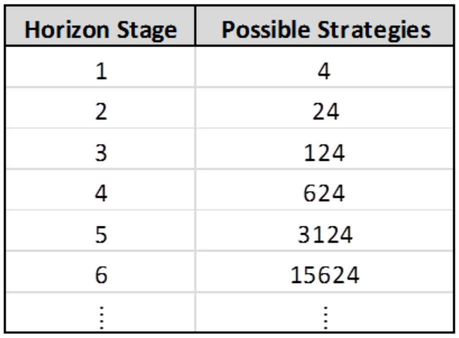
4.2.1 Stationary Information. With a stationary information assumption, information at each observation has the same probability of being correct. For example, past observations in other TIC environments may point to correct classification being 75%, where we assume an observer correctly classifies the target as "legitimate" or "illegitimate" 75% of the time.

In the binary decision for stationary information such as determining the nature of the target, the likelihood of correct classification depends only on the difference in the number of "legitimate" and "illegitimate" observations made at a point in time. For instance, if there are three out of five observations that result in a "legitimate" determination, then the likelihood of this object truly being legitimate is exactly the same as if two out of three observations yield "legitimate" calls. (In this case, both have 0.5 observations above the 50% level.) The likelihood then of correct classification is $p^{2n}/(p^{2n}+q^{2n})$ where n is the number of observations above (or below) 50%. Clearly, as n approaches infinity, then the likelihood of correct classification approaches 1.

Further, we can determine the likelihood of a future observation agreeing with previous observations. In a simple case, assume that $p = 0.75$, what is the likelihood of the second observation agreeing with the first observation? The answer is the likelihood that they are both wrong plus the likelihood they are both correct ($p^2 + q^2 = 0.75^2 + 0.25^2 = 0.625$) and the likelihood that they disagree is then $1 - 0.625 = 0.375$. Moreover, if we know that the previous observations (regardless of how many correct and incorrect observations have been made) yield a 0.75 probability of being correct, then the next observation will agree with the prior consensus 62.5% of the time and disagree 37.5%. We can further prove (see Appendix B) that with information improvement over time, one would never shoot immediately following an "illegitimate" call.

4.2.2 Improving Information. There are reasons to believe that the information gathered in successive stages will improve due to more intelligence assets being placed in the scenario, whether in the form of more ground forces entering the TIC scenario, more UAVs being moved into the environment or more air-to-ground fighters joining the TIC. With this assumption, better information becoming available will typically resolve the nature of the suspected target more quickly.

4.2.3 Recursive Fixing. In the finite horizon scenario for either stationary or improved information, as the horizon stage (the last stage considered) increases linearly the number of possible strategies increases exponentially. For instance, if there is only one stage, then our choices are to "shoot" or "leave" depending on whether the first observation is "legitimate" or "illegitimate". That is, we could have an optimal strategy of (S,S), (S,X), (X,S), or (S,S) for the two possible outcomes of the observation. For the case where there are two observations made, we could have many more strategies since now a strategy after the first observation may be to look "L", giving three possible actions after the first observation. Further, we add another round of observations and decisions when making a second observation and this grows the number of possible strategies exponentially as the stages increase (see Figure 55), note that possible strategies equals 5^{n-1} where n is the horizon stage.



Horizon Stage	Possible Strategies
1	4
2	24
3	124
4	624
5	3124
6	15624
⋮	⋮

Figure 55: Possible Strategies by Horizon

Again, we will assume that information arrives at set intervals each one unit of time apart. The inputs will remain the information distribution as a product of time, the cost of waiting one cycle for more information (c_w), and the cost of striking a building which is not a legitimate military target (c_s). Let's assume, for example, that information follows the cumulative distribution function of the geometric distribution with a probability of p. Thus, the likelihood of getting correct information on the first look is p, on the second look it's $1-(1-p)^2$ and on the third look $1-(1-p)^3$ (that is, on the nth look the likelihood of correct information is $1-(1-p)^n$). Further, we will assume that each look is independent of any other look.

We will assume that our *a priori* assumption is that upon arrival at a TIC, a target is equally likely to be legitimate or illegitimate. Further, if we assume a finite horizon of only one look, then the problem is simple to solve if we assume that a target is equally likely to be legitimate or illegitimate. If upon the first look the target is called "legitimate" then if $\frac{c_w}{c_s} > \frac{1-p}{p}$ the target should be fired upon and if $\frac{c_w}{c_s} < \frac{1-p}{p}$ then no strike should take place. In a similar manner, if the weapon is called "illegitimate" and $\frac{c_s}{c_w} > \frac{1-p}{p}$ then no strike should take place and $\frac{c_s}{c_w} < \frac{1-p}{p}$ indicates that a strike should still take place.

With a finite time horizon and two looks, the problem is more complicated since the two looks may yield different responses. For instance, the first look may call the target "legitimate" and the second calls it "illegitimate". By our assumption the second look is more accurate but is mitigated by the first look yielding a "legitimate" call. For simplicity, call the likelihood of correct information at the nth look $L(p,n)$, where $L(p,n) = 1-(1-p)^n$ and $L'(p,n) = (1-p)^n$ is the likelihood of the nth look being incorrect. If both calls are "legitimate" then the likelihood of the target being legitimate is $\frac{L(p,1)L(p,2)}{L(p,1)L(p,2)+L'(p,1)L'(p,2)}$. If the first call is "legitimate" and the second call is "illegitimate" then the likelihood of the first call being correct is $\frac{L(p,1)L'(p,2)}{L(p,1)L'(p,2)+L'(p,1)L(p,2)}$. Now, we begin to see the dynamic programming formulation of the finite horizon LLS problem. The cost of waiting must not only include c_w, but must also include future expected costs of c_w and c_s.

If we consider "S" and "X" to be the same action (both of which end the situation) then the finite horizon network with two observations looks like:

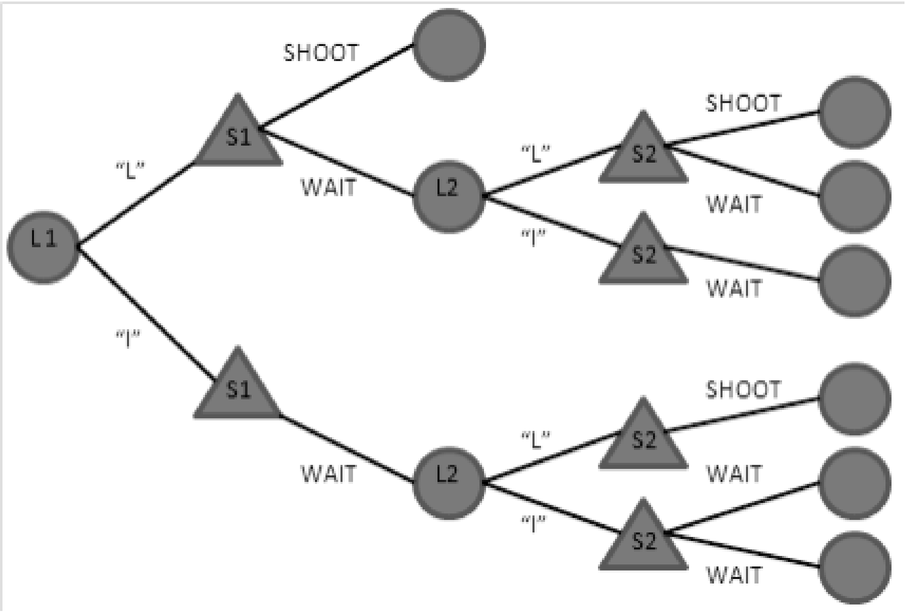


Figure 56: Two-Look Horizon

Again, assume a two-look horizon where a target is equally likely to be legitimate or illegitimate. We can then calculate the likelihood of a “legitimate” or “illegitimate” call at each stage given previous calls. At look 1 (L1) the likelihood of a “legitimate” call is 0.5. The likelihood of L2 being the same as L1 is $p(1-q^2)+q^3$ with the likelihood of being contradictory at $1-(p(1-q^2)+q^3)$. Further, the cost of never shooting equals $2c_w$, the cost of waiting and then shooting being $c_w + p_i c_s$, and the cost of shooting immediately after the first look being $p_i c_s$. While c_s and c_w are constant, p_i (the probability of the target being illegitimate) changes as we get more or different observations. For example, $p_i = q$ if we shoot after L1 returns “legitimate”, however, $p_i = p$ if we shoot after L1 returns “illegitimate”. For the two-horizon problem, shooting after both looks return “legitimate” yields:
$p_i = \frac{L'(p,1)L'(p,2)}{L(p,1)L(p,2)+L'(p,1)L'(p,2)} = \frac{q^3}{q^3+p(1-q^2)}$. Shooting after L1= “I” and L2 = “L” results in:
$p_i = \frac{L'(p,1)L'(p,2)}{L(p,1)L(p,2)+L'(p,1)L'(p,2)} = \frac{pq^2}{q(1-q^2)+pq^2}$.

If we assume $p = 0.75$, then L1 = L2 = “L” yields $p_i = \frac{1}{46}$ whereas if L1 = “I” and L2 = “L” results in $p_i = \frac{1}{6}$. Then, if after two looks, $\frac{c_w}{c_s} > \frac{pq^2}{q(1-q^2)+pq^2}$ then one would shoot after if L2 = “L”, if then one should shoot if L1 = L2 = “L”, and not shoot otherwise. Determining, what to do after the first look is more complicated, since we must incorporate the expected cost after the second look.

If L1 = “L” our cost will be:
$\min\{qc_s, c_w + (p(1-q^w)+q^3)\cdot\min\{c_w, c_s(\frac{q^3}{q^3+p(1-q^2)})\} + c_w(1-p(1-q^2)-q^3)\}$. Using the $p = 0.75$ assumption, this breaks down to $\min\{\frac{c_s}{4}, c_w + \frac{23}{32}\cdot\min\{c_w, \frac{c_s}{46} + \frac{9c_w}{32}\}\}$. If $\frac{c_w}{c_s} > \frac{1}{6}$, then $\min\{\frac{c_s}{4}, c_w + \frac{23c_w}{32} + \frac{9c_w}{32}\} = \min\{\frac{c_s}{4}, 2c_w\} = \frac{c_s}{4}$ meaning we would shoot after L1 = “L”. If $\frac{1}{46} < \frac{c_w}{c_s} < \frac{1}{6}$, then $\min \frac{c_s}{4}, c_w + \frac{23}{32}\cdot\min\{c_w, \frac{c_s}{46}\} + \frac{9c_w}{32} = \min\{\frac{c_s}{4}, c_w + \frac{23}{32}\cdot\frac{c_s}{46} + \frac{9c_w}{32}\} = \min\{\frac{c_s}{4}, \frac{41c_w}{32} + \frac{c_s}{64}\}$, so since $\frac{c_w}{c_s} < \frac{1}{6} < \frac{15}{82}$ then we would wait to shoot.

If L1 = “I”, then our cost will be $c_w + c_w(p(1-q^2)+q^3) + (1-p(1-q^2) - q^3)\cdot\min\{c_w, c_s\frac{pq^2}{q(1-q^2)+pq^s}\}$.

If $\frac{c_w}{c_s} > \frac{1}{6}$, then $c_w + \frac{23c_w}{32} + \frac{9}{32} \cdot \min\{c_w, \frac{c_s}{6}\} = \frac{55c_w}{32} + \frac{9c_s}{192}$. If $\frac{c_w}{c_s} < \frac{1}{6}$, then wait.

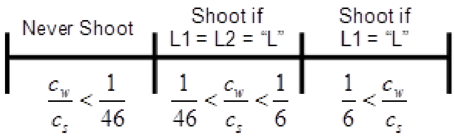

Never Shoot
Shoot if L1 = L2 = "L"
Shoot if L1 = "L"
$\frac{c_w}{c_s} < \frac{1}{46}$
$\frac{1}{46} < \frac{c_w}{c_s} < \frac{1}{6}$
$\frac{1}{6} < \frac{c_w}{c_s}$

Figure 57: Two-Look Horizon Policy ($p = 0.75$)

4.3 Infinite Horizon with Stationary Information

The infinite horizon problem cannot be solved by recursive fixing since there is no final stage. However, for stationary information, we can rely upon the Markov attributes of the problem in order to recursively solve the TIC problem.

4.3.1 Building the Transition Matrix. In the stationary information case, we are able to exploit the Markov nature of the problem since the likelihood of a target being correctly identified as a friend or foe depends only on the difference in the number of "legitimate" and "illegitimate" observations made up to that stage. The state at a given time is identical to the belief state at that time, and not the number of observations made to that point (this is not the case for improving information). Further, we know the likelihood of the next observation being "legitimate" or "illegitimate". Let b be the belief state at a point in time, where b is the greater of the probability that the target is "legitimate" or "illegitimate", then the likelihood that the next observation agrees with the prevailing belief is $b * p + (1 - b) * (1 - p)$. As b grows larger, the likelihood of the next observation agreeing with the prevailing belief increases (with a limit of p).

Therefore, we can make a transition matrix based on the belief state (expressed in terms of the difference between the number of "legitimate" and "illegitimate" calls) prior to the current stage (i.e. P_1 may represent 5 "L"s and 4 "I"s or 1 "L" and 0 "I"s). From P_1, for example, we can only move to P_0 or P_2. The transition matrix, which is infinitely large, becomes:

$$P = \begin{Vmatrix} \ddots & \vdots & \vdots & \vdots & \ddots \\ \cdots & P_{-1,-1} & P_{-1,0} & P_{-1,1} & \cdots \\ \cdots & P_{0,-1} & P_{0,0} & P_{0,1} & \cdots \\ \cdots & P_{1,-1} & P_{1,0} & P_{1,1} & \cdots \\ \ddots & \vdots & \vdots & \vdots & \ddots \end{Vmatrix} \tag{29}$$

with $\sum_{j=-\infty}^{\infty} P_{i,j} = 1, i \in \mathbb{Z}$ and $P_{i,j} = 0$ if $|i-j| \neq 1$.

If we choose to truncate the matrix at a given point (in this case, looking from i and j from -3 to 3) and we insert the transition probabilities then the result is:

$$\begin{Vmatrix} \approx 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{p^3+q^3}{p^2+q^2} & 0 & \frac{p^2q+pq^2}{p^2+q^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{p^2+q^2}{p+q} & 0 & \frac{2pq}{p+q} & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & \frac{2pq}{p+q} & 0 & \frac{p^2+q^2}{p+q} & 0 \\ 0 & 0 & 0 & 0 & \frac{p^2q+pq^2}{p^2+q^2} & 0 & \frac{p^3+q^3}{p^2+q^2} \\ 0 & 0 & 0 & 0 & 0 & 0 & \approx 1 \end{Vmatrix} \tag{30}$$

Since $P_{i,j} = P_{-i,-j}$ we can convert the transition matrix to:

$$P = \begin{Vmatrix} 0 & P_{0,1} & 0 & 0 & 0 & \cdots \\ P_{1,0} & 0 & P_{1,2} & 0 & 0 & \cdots \\ 0 & P_{2,1} & 0 & P_{2,3} & 0 & \cdots \\ 0 & 0 & P_{3,2} & 0 & P_{3,4} & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots \end{Vmatrix} \tag{31}$$

Now that we have described the transition matrix we can use it to guide the optimal strategy determination. If the likelihood of incorrectly identifying the target as a legitimate target is below the cost of waiting divided by the cost of shooting at an illegitimate target then we would always fire. That is, if $p_i = 1 - p_l < c_w/c_s$ then we would fire upon the target.

By looking at the limiting behavior of the transition matrix ($P^{(\infty)}$), we then can see the likelihood of the target truly being legitimate or illegitimate based on the previous observations. Again, if we had only one observation and it was "legitimate" then the likelihood of that target truly being legitimate is 0.75 (the same is true if we had two "legitimate" calls and one "illegitimate" call). Thus the likelihood of being correct based on the difference in the number of observations is

$$\begin{bmatrix} \cdots & \frac{p^2}{p^2+q^2} & \frac{p}{p+q} & 0.5 & \frac{p}{p+q} & \frac{p^2}{p^2+q^2} & \cdots \end{bmatrix} \tag{32}$$

$$P^{(\infty)} = \begin{Vmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{p^2}{p^2+q^2} & 0 & \cdots & 0 & \frac{q^2}{p^2+q^2} \\ \frac{p}{p+q} & 0 & \cdots & 0 & \frac{q}{p+q} \\ 0.5 & 0 & \cdots & 0 & 0.5 \\ \frac{q}{p+q} & 0 & \cdots & 0 & \frac{p}{p+q} \\ \frac{q^2}{p^2+q^2} & 0 & \cdots & 0 & \frac{p^2}{p^2+q^2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{Vmatrix} \tag{33}$$

If we cut off the number of observations above 50% necessary to make a definitive call then the transition matrix becomes finite with two absorbing states (one for targets deemed "illegitimate" and one for those deemed "legitimate"). This adjustment is supported by $p_i = 1 - p_l < \frac{c_w}{c_s}$ determination, which prescribes that if $\frac{q_i}{p_i+q_i} < \frac{c_w}{c_s}$ then we will fire, thus negating the need for further observations. Once we find the i for which "S" or "X" (i.e. $\frac{q^i}{p^i+q^i} < \frac{c_w}{c_s}$) is the optimal policy then we can recursively find the optimal strategy for any i.

4.3.2 Mean Time Spent in Transient States. Assuming we have found the terminating state i, then we will determine the optimal policy recursively for state $i-1$, $i-2$, and so on until reaching 0. To accomplish this we must know the relative costs of actions in state $i-1$. In state $i-1$, we will know the cost associated with the "S" or "X" action in state i, but we need to know the expected number of observations necessary to reach state i from $i-1$ (or the mean time spent in transient states $(0, 1, \ldots, i-1)$) [85].

$$\mathbf{P}_T = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1t} \\ \vdots & \vdots & \vdots & \vdots \\ P_{t1} & P_{t2} & \cdots & P_{tt} \end{bmatrix} \tag{34}$$

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1t} \\ \vdots & \vdots & \vdots & \vdots \\ s_{t1} & s_{t2} & \cdots & s_{tt} \end{bmatrix} \tag{35}$$

$$\begin{aligned} \mathbf{S} &= \mathbf{I} + \mathbf{P}_T\mathbf{S} \qquad (36) \\ (\mathbf{I} - \mathbf{P}_T)\mathbf{S} &= \mathbf{I} \\ \mathbf{S} &= (\mathbf{I} - \mathbf{P}_T)^{-1} \end{aligned}$$

Assuming $i = 5$, $p = 0.75$, then our S matrix would be:

$$\mathbf{S} = (\mathbf{I} - \mathbf{P}_T)^{-1} = \begin{bmatrix} 1.98 & 2.62 & 2.13 & 1.84 & 1.34 \\ 0.98 & 2.62 & 2.13 & 1.84 & 1.34 \\ 0.38 & 1.02 & 2.13 & 1.84 & 1.34 \\ 0.13 & 0.34 & 0.70 & 1.84 & 1.34 \\ 0.03 & 0.09 & 0.18 & 0.47 & 1.34 \end{bmatrix} \tag{37}$$

The S matrix shows that if we are in state $i - 1 = 4$, then we will need $0.03 + 0.09 + 0.18 + 0.47 + 1.34 = 2.11$ more observations on average to reach the absorbing state i where we know that the optimal strategy is “X” or “S”.

4.3.3 Constructing the Optimal Policy. Let $s_i = \sum_{n=1}^{t} s_{in}$ which indicates the expected number of transitions from the ith state until reaching the terminating state. If $s_i c_w > \frac{q^i}{p^i+q^i} c_s$ then the optimal policy would be to shoot upon reaching state i. Further, since there is uncertainty still at the terminating state equal to $\frac{p^{t+1}}{p^{t+1}+q^{t+1}}$ then if $s_i c_w + \frac{q^{t+1}}{p^{t+1}+q^{t+1}} c_s > \frac{q^i}{p^i+q^i} c_s$ the optimal policy is to shoot, in this example $p = 0.75$, and the ratio of c_w to c_s is 0.01.

The infinite horizon scenario for the LLS can be mapped to the tiger problem presented in [50], where the observer must choose between continuing to listen for a tiger (at a small cost) or open a door revealing a tiger (or fortune) at a large cost or reward. Once the observer’s belief state reaches a certain level of confidence, then he would choose to open the door. This problem has a fairly simple structure for the optimal decision policy, where the three choices are to open the left door, continue to listen, and open the right door, are three non-overlapping segments of the [0,1] belief state. As a reminder, though there may be only three possible decisions in an POMDP, the number of ranges where one decision is optimal is infinite. The challenge then is to determine the two points in the belief state where we move from “leave” to “look” and from “look” to “shoot”. In the case where the cost of firing upon an invalid target equals the cost of leaving when the target is valid, then the two points are equal distances from 0 and 1 (i.e. the two points add to 1).

4.4 Results

Now that we can construct the optimal policy in the infinite horizon, stationary information problem, we investigate the effects of the parameters of the problem on the average time to make decisions and the average cost of a TIC situation.

4.4.1 Effects of Intelligence on Decisions. In Figure 58 we view the effects of intelligence accuracy on the observation difference necessary to make an "S" or "X" decision. In this example, we assume that c_w/c_s is 0.01. We see that, as one might expect, extremely accurate intelligence leads to a very small number of observations necessary to make an "S" or "X" decision (in the case where $p = 0.99$, we would immediately make an "S" or "X" determination). However, of interest, is that extremely poor information, such as $p = 0.5$, results in a similar choice of "S" or "X" after the first observation. This basically tells us that our information is so unreliable, that waiting for more "bad" information will do us no good. We will have no improved belief state with time.

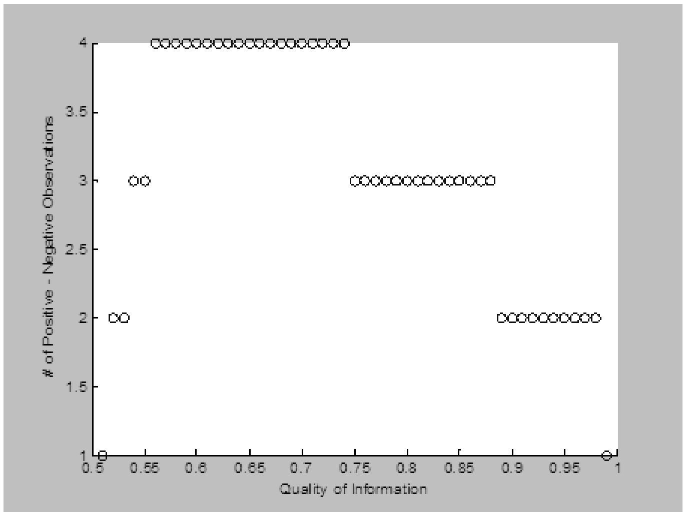


Figure 58: Quality of Information vs. Observations

4.4.2 Effects of a priori Information on Decisions. Up to this point, the assumption has been that the observer is equally likely to see either a valid or invalid target. However, if, based on experience, historical evidence points to a different ratio of valid to invalid targets then we can incorporate this into the model. *A priori* information will have the same effect as previous observations would have. *A priori* information will give the initial belief state of the system, whereas before, the initial belief state was $(0.5, 0.5)$, if the accuracy of the *a priori* information is p_p, then our initial belief state will be $(p_p, 1-p_p)$ or $(1-p_p, p_p)$. The same technique of finding the absorbing state and recursively setting the optimal policy will apply. The transition matrix will reflect the *a priori* information where the entries to the left and right of the diagonal are multiplied by $2p_p$ and $2(1-p_p)$ or vice versa.

4.4.3 Effects of Weights on Decisions. In Figure 59, we vary the ratio of c_w to c_s from 0.1 to 1.0 to see the effect of the relative costs on our decision threshold. As the cost of waiting decreases relative to the cost of either firing on an illegitimate target or not firing on a legitimate target the number of observations necessary to make a “S” or “X” determination increases.

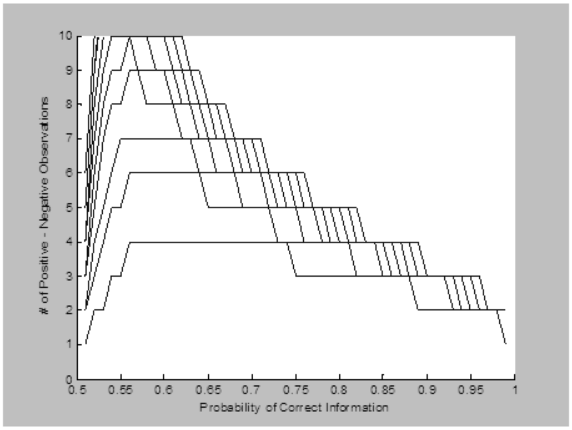


Figure 59: Effect of Changing Costs

Further, we have assumed that the cost of shooting and the cost of leaving (reaching the conclusion that the observer no longer waits) were equal. However, if the cost of shooting and the cost of leaving were unequal, then the transition matrix would have to be altered. Additionally, this would create two different levels for the two absorbing states (the points where the observer would definitely leave the scene or definitely fire upon the target).

The findings of this chapter are based on a multitude of assumptions in an effort to keep this material unclassified. However, the sensitivity analyses provide understanding of the various factors at play in these TIC situations.

4.5 Conclusion and Future Work

In this chapter, we have provided a framework for making optimal policy decisions in fast-moving TIC situations where observers are unsure of the nature of possible enemy forces in both finite horizon and infinite horizon problems. Through the recursive technique of solving this Markov decision process we have demonstrated the effect of improved intelligence and differing weights concerning waiting and making incorrect decisions in the face of uncertain situations.

Future work involves creating heuristics for solving the TIC problem for the infinite horizon with improving information. In these situations, the Markov property will not hold limiting the ability to apply many of the techniques in this paper. Additionally, making the problem more real-world reflective will lead to more complicated cost and decision parameters.

V. Summary, Future Work, and Conclusions

5.1 Summary of Original Contribution

In this dissertation, a characterization of the distribution of supply airdrops and methods for optimally dropping them is presented. Specifically, supply airdrops follow a bivariate normal distribution in which the x and y deviations are uncorrelated ($\rho = 0$). A surrogate approximation function for the bivariate normal distribution supports quick integration of the distribution to assess drop risk. RSM with surrogate, and DE, both return Pareto optimum results depending on a tradeoff between runtime and accuracy. Both achieve near-optimal solutions of the non-linear program resulting from the airdrop problem, quickly finding settings for both airdrop location and approach angle. Enumeration is strongly dominated by all other algorithms.

We suppose an airdrop planner who has been shown the oval shapes and scales of a bundled set of a supply airdrop could predict the optimal aimpoint within 50 meters in each direction and the drop angle within 5 degrees angle of the optimal solution. Note that this is a high standard - we have looked at hundreds of combinations of drops, yet still only approach that level of accuracy. In our base problem, where the collateral objects have the same weighting, the planner "eyeballing" a solution would have a collateral risk 14% higher than the optimum. In more complicated scenarios where the collateral objects are weighted differently, "eyeballing" a solution becomes much worse than the solutions found by our algorithms, with "eyeballed" solutions routinely worse by 20% or more. A more reliable technique must be implemented to limit damage and ensure recoverability.

Additionally, a quick and accurate algorithm for accurately creating the Pareto optimal frontier in the multi-objective airstrike problem is presented. This algorithm, which leverages specific attributes of lethality and collateral risk, is shown to routinely outperform differential evolution and enumeration algorithms. Once Pareto optimal solutions are found these can be quickly converted to solutions to the associated goal-programming or weighted sum scalarization problems. The choice of damage function is shown to greatly affect the expected lethality and collateral risk in an airstrike underscoring the need for accurate estimation of weapons effects.

We demonstrate that the current methodology of not using offset aiming yields lethality 26% higher at a cost of collateral risk 176% higher than a collateral first approach. The algorithm presented can be incorporated into the weapon (and employment) decisions facing an airstrike planner, who could alter selections based on the minimum lethality needed or maximum collateral risk allowed to remedy the limitation from non-offset targeting.

Finally, we provide a framework for making optimal policy decisions in fast-moving TIC situations where observers are unsure of the nature of possible enemy forces in both finite horizon and infinite horizon problems. Through the recursive technique of solving this Markov decision process we have demonstrated the effect of improved intelligence and differing weights concerning waiting and making incorrect decisions in the face of uncertain situations.

5.2 Future Work

The future work for the research presented in this dissertation will be modifying the algorithms and theory to real-world software and application. Tools currently in use by the USAF have more complicated inputs for both airdrop and airstrike collateral estimates. While the tools being implemented today provide more accuracy than the assumptions in this work, they all seem to lack the optimization step that is necessary to truly lower collateral risk.

5.3 Conclusions

The importance of collateral damage minimization in U.S. engagements around the world is undeniable. While steps have been taken to estimate collateral risks for airdrop and airstrike missions, there has been little done to minimize this collateral risk efficiently. For airdrops, there is no tool available to find optimal locations within a drop area to avoid collateral risk while ensuring recoverability, typically, trained mission planners look for areas within a scene to make a drop. Results from this work indicate that planners could be greatly aided by the work presented.

For airstrikes, offset aiming is a vital piece of mission planning, and one that should be incorporated in the earliest stages of collateral damage estimation. Further, the cookie-cutter damage function should be scrapped in favor of more representative damage functions. While these functions may be more difficult to visualize, the software packages available to mission planners should have no issues with handling the more complicated distributions.

TIC scenarios present the greatest collateral risk and the most difficult type of risk to lower. Improved intelligence gathering and an *a priori* understanding of tradeoffs within a TIC have been shown to speed up decision-making in these time-sensitive engagements.

Collateral damage and civilian deaths continue to plague U.S. missions worldwide and attempts to minimize these risks are at the forefront of military leaders' efforts. This dissertation presents important improvements in understanding the source of collateral risk and steps which the U.S. military can take to minimize risk while still ensuring mission success.

Appendix A. Proof 1

THEOREM: The expected collateral risk in a randomly-generated infinitely-large scene $E[f_{2_{cc}}] \leq E[f_{2_g}] \leq E[f_{2_e}]$ regardless of the accuracy and lethal range of the weapon.

PROOF: Expected collateral risk is independent of accuracy when randomly aiming by Formula 18. Additionally, by assumption, the lethal range for each damage function is equal, i.e. $LR = \int_0^\infty d_{cc}(r)dr = LR = \int_0^\infty d_g(r)dr = LR = \int_0^\infty d_e(r)dr$.

Each pair-wise set of damage functions overlaps only once. For the cookie-cutter and the Gaussian damage function, the functions only intersect at $r = LR$ since $r < LR \rightarrow d_{cc} = 1$ and $r > LR \rightarrow d_{cc} = 0$ and $0 < d_g < 1$, when $r < LR$ then $d_{cc} > d_g$ when $r > LR$ then $d_{cc} < d_g$. For the Gaussian and exponential functions, the two functions cross only at the point $r = \frac{4LR}{\pi}$ with $d_e < d_g$ for $r < \frac{4LR}{\pi}$ and $d_e > d_g$ for $r > \frac{4LR}{\pi}$.

LEMMA: If $\int_0^\infty d_1(r)dr = \int_0^\infty d_2(r)dr$, $d_1(r) < d_2(r)$ for $r \in [0, x)$, and $d_1(r) > d_2(r)$ for $r \in (x, \infty)$ then $\int_0^\infty r d_1(r)dr \geq \int_0^\infty r d_2(r)dr$.

$$\begin{aligned}
\int_0^\infty d_1(r)dr &= \int_0^\infty d_2(r)dr \\
\int_0^x d_1(r)dr + \int_x^\infty d_1(r)dr &= \int_0^x d_2(r)dr + \int_x^\infty d_2(r)dr \\
\int_x^\infty d_1(r)dr - \int_x^\infty d_2(r)dr &= \int_0^x d_2(r)dr - \int_0^x d_1(r)dr \\
\int_x^\infty [d_1(r) - d_2(r)]dr &= \int_0^x [d_2(r) - d_1(r)]dr \\
x\int_x^\infty [d_1(r) - d_2(r)]dr &= x\int_0^x [d_2(r) - d_1(r)]dr \\
\int_x^\infty x[d_1(r) - d_2(r)]dr &= \int_0^x x[d_2(r) - d_1(r)]dr \\
\int_x^\infty r[d_1(r) - d_2(r)]dr \geq \int_x^\infty x[d_1(r) - d_2(r)]dr &= \int_0^x x[d_2(r) - d_1(r)]dr \geq \int_0^x r[d_2(r) - d_1(r)]dr \\
\int_x^\infty r[d_1(r) - d_2(r)]dr &\geq \int_0^x r[d_2(r) - d_1(r)]dr \\
\int_x^\infty rd_1(r)dr - \int_x^\infty rd_2(r)dr &\geq \int_0^x rd_2(r)dr - \int_0^x rd_1(r)dr \\
\int_x^\infty rd_1(r)dr + \int_0^x rd_1(r)dr &\geq \int_0^x rd_2(r)dr + \int_x^\infty rd_2(r)dr \\
\int_0^\infty rd_1(r)dr &\geq \int_0^\infty rd_2(r)dr
\end{aligned}$$

Therefore, $\int_0^\infty rd_{cc}(r)dr \leq \int_0^\infty rd_g(r)dr \leq \int_0^\infty rd_c(r)dr \rightarrow 2\pi n\int_0^\infty rd_{cc}(r)dr \leq 2\pi n\int_0^\infty rd_g(r)dr \leq 2\pi n\int_0^\infty rd_c(r)dr$, thus $E[f_{2_{cc}}] \leq E[f_{2_g}] \leq E[f_{2_c}]$.

Appendix B. Proof 2

THEOREM: No optimal policy recommends firing after receiving an "illegitimate" call when $p > 0.5$.

PROOF:

CASE I: Previous to the "illegitimate" call, there have been more "legitimate" calls than "illegitimate" calls.

The "illegitimate" call would move the Markov decision process to a state already visited in the scenario. Due to the Markov property, only the state currently in (and not the path to that state) determines the policy for that state. If the optimal policy at the new state is "shoot", then the observer would have already shot when at the state previously.

CASE II: Previous to the "illegitimate" call, there have not been more "legitimate" calls than "illegitimate" calls.

The belief state (legitimate, illegitimate) is $(\frac{q^i}{p^i+q^i}, \frac{p^i}{p^i+q^i})$ when there have been i more "illegitimate" calls than "legitimate" calls, with $i \geq 0$. With the likelihood of a correct observation $p > 0$, then $\frac{p^i}{p^i+q^i} > \frac{q^i}{p^i+q^i}$ meaning the likelihood of the target being illegitimate is greater than the likelihood of the target being legitimate. Since the cost of leaving a legitimate target equals the cost of firing at a illegitimate target, then the cost of leaving the scenario $(\frac{q^i}{p^i+q^i}c_s)$ is lower (more optimal) than the cost of firing upon the target $(\frac{p^i}{p^i+q^i}c_s)$.

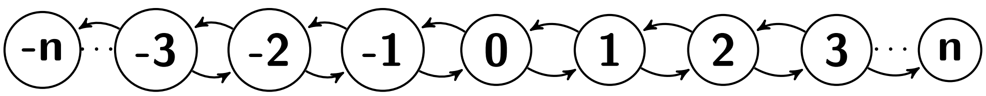


Figure 60: Markov Transition Diagram

Bibliography

1. Abawi, A., N. Robertson, C. Lawrence, and B. Starr. "U.S. launches 'major operation' in Afghanistan". *CNN.com*, July 2 2009.

2. Abbass, H.A., R. Sarker, and C. Newton. "PDE: A Pareto-frontier Differential Evolution Apporach for Multi-objective Problems". *Evolutionary Computation*, (2):971–978, 2001.

3. Ahner, D.K. *Planning and Control of Unmanned Aerial Vehicles in a Dynamic Stochastic System.* PhD dissertation, Boston Univeristy, 2005.

4. Anderson, R.L. "Recent Advances in Finding Best Operating Conditions". *Journal of American Statistical Association*, 48(264):789–798, 1953.

5. Andrews, R. *Airdrop Targeting.* Technical report, United States Air Force (AMC/A2), 2010.

6. Archetti, F. and F. Schoen. "A Survey on the Global Optimization Problem: General Theory and Computational Approaches". *Annals of Operations Research*, 1(2):87–110, 1984.

7. Arkin, W.M. "Smart Bombs, Dumb Targeting?" *Bulletin of the Atomic Scientists*, 56(3):46–53, 2000.

8. Avriel, M. and A.C. Williams. "The Value of Information and Stochastic Programming". *Operations Research*, 18(5):947–954, 1970.

9. Babu, B. and M.M.L. Jehan. "Differential Evolution for Multi-Objective Optimization". *Evolutionary Computation*, 4(11):2696–2703, 2003.

10. Back, T., F. Hoffmeister, and H.-P. Schwefel. "A survey of evolution strategies". *Proceedings of the 4th International Conference on Genetic Algorithms*, 2–9, 1991.

11. Bazaraa, M.S., H.D. Sherali, and C.M. Shetty. *Nonlinear Programming.* Wiley and Sons, 2006.

12. Benney, R., J. Barber, J. McGrath, J. McHugh, G. Noetscher, and G. Tavan. "The joint precision airdrop system advanced concept technology demonstration". *18th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, 2005.

13. Bertsekas, D.P. *Dynamic Programming and Optimal Control.* Athena Scientific, 2005.

14. Binninger, G.C. *Collateral Damage Probability Models.* Technical report, Defense Nuclear Agency, 1980.

15. Birge, J.R. "Stochastic Programming Computation and Applications". *INFORMS Journal on Computing*, 9(2):111–133, 1997.

16. Birge, J.R. and D.F. Holmes. "Efficient Solution of Two-Stage Stochastic Linear Programs Using Interior Point Methods". *Computational Optimization and Applications*, 1(3):245–276, 1992.

17. Birge, J.R. and L. Qi. "Computing Block-Angular Karmarkar Projections with Applications to Stochastic Programming". *Management Science*, 34(12):1472–1479, 1988.

18. Blomvall, J. and P.O. Lindberg. "A Riccati-based primal interior point solver for multistage stochastic programming". *European Journal of Operational Research*, 143(2):452–461, 2001.

19. Branke, J., K. Deb, K. Miettinen, and R. Slowinski. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer, 2008.

20. Brooks, H., T. DeKeyser, D. Jaskot, D. Sibert, R. Sledd, W. Stilwell, and W. Scherer. "Using Agent-Based Simulation to Reduce Collateral Damage during Military Operations". *Proceedings of the 2004 Systems and Information Engineering Design Symposium*, 71–78, 2004.

21. Brooks, S.H. "A Comparison of Maximum-Seeking Methods". *Operations Research*, 7(4):430–457, 1958.

22. Broyles, J.R. *Markovian Models of Patient Throughput in Hospitals: A Regression and Decision Process Approach*. PhD dissertation, Arizona State University, 2009.

23. Cammarano, V.R. *Estimating Cargo Airdrop Collateral Damage Risk*. Master's thesis, Air Force Institute of Technology, 2011.

24. CBSNews. "U.S. Military Begins Air Drops in Haiti". *CBSNews.com*, January 18, 2010.

25. Chairman of the Joint Chiefs of Staff. *Chairman of the Joint Chiefs of Staff Instruction 3160.01*. Technical report, Department of Defense, 2009.

26. challenge.gov. "Humanitarian Air Drop". *challenge.gov*, 2011.

27. Coles, J.B., J. Zhuang, and J. Yates. "Case study in disaster relief: A descriptive analysis of agency partnerships in the aftermath of the January 12th, 2010 Haitian earthquake". *Socio-Economic Planning Sciences*, 46(1):67–77, 2011.

28. Dantzig, G.B. and A. Madanksy. "On the Solution of Two-Stage Linear Programs under Uncertainty". *Proceedings of the Fourth Berkeley Symposium*, 1961.

29. David, I. "Safe Distances". *Naval Research Logistics*, 48(4):259–269, 2001.

30. Davis, L. "Adapting operator probabilities in genetic algorithms". *Proceedings of the 3rd International Conference on Genetic Algorithms*, 61–69, 1989.

31. Denardo, E.V. *Dynamic Programming*. Dover Publications, Inc., 2003.

32. Dillenburger, S.P. *Minimization of Collateral Damage in Airdrops and Airstrikes*. PhD dissertation, Air Force Institute of Technology, 2012.

33. Douglass, R. "Military Robotics and Collateral Damage". *Command and Control Research and Technology Symposium*, 2004.

34. Driels, M.R. *Weaponeering: Conventional Weapon System Effectiveness*. American Institute of Aeronautics and Astronautics, 2004.

35. Eiben, A., P.-E. Raue, and Z. Ruttkay. "Genetic algorithms with multi-parent recombination". *Proceedings of the 3rd Conference on Parallel Problems Solving from Nature*, 78–87, 1994.

36. Fenrick, W.J. "Targeting and Proportionality during the NATO Bombing Campaign against Yugoslavia". *European Journal of International Law*, 12(3):489–502, 2001.

37. Fogel, D. and L. Stayton. "On the effectiveness of crossover in simulated evolutionary algorithms". *BioSystems*, (32):171–182, 1994.

38. Garlasco, M. *Troops in Contact: Airstrikes and Civilian Deaths in Afghanistan*. Technical report, Human Rights Watch, 2008.

39. Garlasco, M. "How to Cut Collateral Damage in Afghanistan". *Strategies of Development and Transformation*, March 3, 2010.

40. Gassmann, H. "MSLiP: A Computer Code for the Multistage Stochastic Linear Programming Problem". *Mathematical Programming*, 47(1-3):407–423, 1990.

41. Glazebrook, K. and A. Washburn. "Shoot-Look-Shoot: A Review and Extension". *Operations Research*, 52(3):454–463, 2004.

42. Gosavi, A. *Simulation-Based Optimization*. Kluwer Academic Publishers, 2003.

43. Graham, B. "Military Turns to Software to Cut Civilian Casualties". *Washington Post*, February 21, 2003.

44. Hart, W. *Adaptive Global Optimization with Local Search*. PhD dissertation, University of California, San Diego, 1994.

45. Herold, M.W. *A Dossier on Civilian Victims of United States' Aerial Bombing of Afghanistan: A Comprehensive Accounting*. Technical report, University of New Hampshire, 2001.

46. Higle, J.L. and S. Sen. "Stochastic Decomposition: An Algorithm for Two-Stage Linear Programs with Recourse". *Mathematics of Operations Research*, 16(3):650–669, 1991.

47. Holden, M. “Afghan girl killed by British leaflet drop”. *Reuters*, September 30, 2009.

48. Huang, V.L., A.K. Qin, and P.N. Suganthan. “Self-adaptive Differential Evolution Algorithm for Constrained Real-Parameter Optimization”. *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, 17–24, 2006.

49. Humphrey, A., J. See, and D. Faulkner. “A Methodology to Assess Lethality and Collateral Damage for Nonfragmenting Precision-Guided Weapons”. *International Test and Evaluation Association*, 29:411–419, 2008.

50. Kaelbling, L.P., M.L. Littman, and A.R. Cassandra. “Planning and acting in partially observable stochastic domains”. *Artificial Intelligence*, 101(1-2):99–134, 1998.

51. Kall, P. and S.W. Wallace. *Stochastic Programming*. John Wiley and Sons, 1994.

52. Karmarkar, N.K. “A New Polynomial-Time Algorithm for Linear Programming”. *Combinatorica*, 4:373–395, 1984.

53. Karnopp, D.C. “Random Search Techniques for Optimization Problems”. *Automatica*, 1963.

54. Keaney, T. *Collateral Damage in the Gulf War: Experience and Lessons*. Technical report, The Carr Center for Human Rights Policy, 2002.

55. Kiernan, B. and T. Owen. “Roots of U. S. Troubles in Afghanistan: Civilian Bombing Casualties and the Cambodian Precedent”. *The Asia-Pacific Journal*, June 28, 2010.

56. Kocher, M.G., J. Pahlke, and S.T. Trautmann. *Tempus Fugit: Time Pressure in Risky Decisions*. Technical report, University of Munich, 2011.

57. La Rock, H.L. *Decision Criteria for the Use of Cannon-Fired Precision Munitions*. Master’s thesis, Naval Postgraduate School, 2005.

58. Lampinen, J. “A Constraint Handling Approach for the Differential Evolution Algorithm”. *Proceedings of the Evolutionary Computation*, 2:1468–1473, 2002.

59. Lampinen, J. and I. Zelinka. “Mixed Integer-Discrete-Continuous Optimization by Differential Evolution”. *Proceedings of MENDEL’99*, 71–76, 1999.

60. Larter, D. “War-zone airdrops reach record-breaking pace”. *Air Force Times*, May 28, 2011.

61. Lasdon, L. *Optimization Theory for Large Systems*. MacMillan Publishing Co., 1970.

62. Lin, Y.-H., R. Batta, P.A. Rogerson, A. Blatt, and M. Flanigan. “A logistics model for emergency supply of critical items in the aftermath of a disaster”. *Socio-Economic Planning Sciences*, 45(4):132–145, 2011.

63. Lipina, A. and F. Schmokel. *Improved Common Understanding is Critical to Reducing Civilian Casualties in Afghanistan*. Technical report, United States Department of Defense, 2010.

64. Lucas, T.W. “Damage Functions and Estimates of Fratricide and Collateral Damage”. *Naval Research Logistics*, 50(4):306–321, 2003.

65. Lustig, I., J. Mulvey, and T. Carpenter. “Formulating Two-Stage Stochastic Programs for Interior Point Methods”. *Operations Research*, 39(5):757–770, 1991.

66. Mail Foreign Service. “Haiti earthquake disaster: U.S. resorts to ‘risky’ airdrops as transport bottlenecks keep aid from desperate survivors”. *Dailymail.co.uk*, January 19, 2010.

67. Mays, G.C. and P.D. Smith. *Blast Effects on Buildings*. Thomas Telford Publications, 1995.

68. McAllister, D.B. *Planning with Imperfect Information*. Technical report, Massachusetts Institute of Technology, 2006.

69. McGowan, L. “Airdrop system resupplies ground troops”. *Transformation*, August 16, 2006.

70. Michalewicz, Z. and M. Schoenauer. “Evolutionary Algorithms for Constrained Parameter Optimization Problems”. *Evolutionary Computation*, 4(1):1–32, 1996.

71. Mills, C.A. “The design of concrete structure to resist explosions and weapon effects”. *Proceeding of the 1st International Conference on concrete for hazard protections*, 61–73, 1987.

72. Monahan, G.E. “Optimal stopping in a partially observable Markov process with costly information”. *Operations Research*, 28(6):1319–1334, 1980.

73. Monahan, G.E. “A Survey of Partially Observable Markov Decision Markov Decision Processes: Theory, Models, and Algorithms”. *Management Science*, 28(1):1–16, 1982.

74. Myers, R.H. and D.C. Montgomery. *Response Surface Methodology*. John Wiley and Sons, Inc., 2002.

75. Myers, S.L. “Chinese Embassy Bombing: A Wide Net of Blame”. *The New York Times*, April 17, 2000.

76. Newmark, N.M. and R.J. Hansen. “Design of blast resistant structures”. *Shock and Vibration Handbook*, 3, 1961.

77. Ngo, T., P. Mendis, A. Gupta, and J. Ramsay. “Blast Loading and Blast Effects on Structures - An Overview”. *Electronic Journal of Structural Engineering*, 2007.

78. Niederreiter, H. *Random Number Generation and quasi-Monte Carlo Methods.* Society for Industrial and Applied Mathematics, 1992.

79. Payne, J.W. and J.R. Bettman. "Where Time is Money: Decision Behavior under Opportunity-Cost Time Pressure". *Organizational Behavior and Human Decision Processes*, 66(2):131–152, 1996.

80. Phillips, D.D., W. Kang, D.K. Ahner, and B.K. Mansager. "A Dynamic System Model of Information Flow on the Battlefield". *IEEE International Conference on Systems, Man, and Cybernetics*, 685–690, 2007.

81. Polikar, R. "Ensemble Based Systems in Decision Making". *IEEE Circuits and Systems Magazine*, (3):21–45, 2006.

82. Przemieniecki, J. *Mathematical Models in Defense Analyses.* American Institute of Aeronautics and Astronautics, 2000.

83. Qiu, Q. and M. Pedram. *Dynamic Power Management Based on Continuous-Time Markov Decision Process.* Technical report, University of Southern California, 1999.

84. Rastrigin, L.A. "The Convergence of the Random Search Method in the Extremal Control of a Many-Parameter System". *Automatic Remote Control*, 1963.

85. Ross, S.M. *Probability Models.* Academic Press, 2007.

86. Roy, A., L. Lasdon, and D. Plane. "End-User Optimization with Spreadsheet Models". *European Journal of Operational Research*, 39(2):131–137, 1989.

87. Schwefel, H.-P. *Numerical Optimization of Computer Models.* John Wiley and Sons, 1981.

88. Science Applications, Inc. *Collateral Damage Probability Models.* Technical report, Defense Nuclear Agency, 1980.

89. Shah, A. "Afghan leader: airstrikes could undermine US pact". *Daily Star*, May 8, 2012.

90. Shanker, T. "Joint Chiefs Chairman Readjusts Principles on Use of Force". *New York Times*, March 3, 2010.

91. Shaughnessy, L. "U.S. Air Force drops 55,000 pounds of food, water into Haiti". *CNN.com*, January 18, 2010.

92. Smallwood, R. D. and E. J. Sondik. "The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon". *Operations Research*, 21(5):1071–1088, 1973.

93. Solis, F.J. and R.J-B. Wets. "Minimization by Random Search Techniques". *Mathematics of Operations Research*, 6(1):19–30, 1981.

94. Storn, R. and K. Price. "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces". *Journal of Global Optimization*, 11(4):341–359, 1997.

95. Strazicky, B. "Some Results Concerning an Algorithm for the Discrete Recourse Problem". *Stochastic Programming*, 263–271. Academic Press, 1980.

96. Times, New York. "Costly collateral damage in Afghanistan". *New York Times*, June 21, 2007.

97. Tyson, A.S. "Top U.S. Commander in Afghanistan Is Fired". *Washington Post*, May 12, 2009.

98. United Nations Assistance Mission Afghanistan. *Afghanistan, Annual Report 2010, Protection of Civilians in Armed Conflict*. Technical report, United Nations Assistance Mission in Afghanistan, 2011.

99. United States Air Force. *AFI 11-231, Computed Air Release Point Procedures*. Technical report, United States Air Force, 2005.

100. United States Air Force. *Airdrop Collateral Damage Estimation Methodology*. Technical report, United States Air Force, 2010.

101. United States Army. *TM 5-1300: The Design of Structures to Resist the Effects of Accidental Explosions*. Technical report, US Department of the Army, Navy, and Air Force, 1990.

102. Van Slyke, R. and R. Wets. "L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming". *SIAM Journal of Applied Mathematics*, 17(4):638–663, 1969.

103. Wafa, A.W. and J.F. Burns. "U.S. Airstrike Reported to Hit Afghan Wedding". *New York Times*, November 6, 2008.

104. Xue, F., A.C. Sanderson, and R.J. Graves. "Pareto-based Multi-Objective Evolution". *The 2003 Congress on Evolutionary Computation*, (2):862–869, 2003.

105. Yost, K.A. and A.R. Washburn. "The LP/POMDP Marriage: Optimization with Imperfect Information". *Naval Research Logistics*, 47:607–619, 2000.

106. Zitzler, E., K. Deb, and L. Thiele. "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results". Institut fur Technische und Kommunikationsnetze, Zurich.

107. Zitzler, E. and L. Thiele. "Multiobjective Evolutionary Algorithms: A comparative case study and strength pareto approach". *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Vita

Major Steven P. Dillenburger graduated from the University of Notre Dame in May 2002 with a Bachelor of Science degree in Mathematics and a Bachelor of Business Administration degree in Accounting. He was commissioned an officer in the United States Air Force through the Reserve Officers Training Corps upon graduation in 2002.

His first assignment was to the Air Force Research Laboratory Sensors Directorate at Wright-Patterson Air Force Base (WPAFB), Ohio. While stationed at WPAFB, he completed his Master of Science in Mathematics from Wright State University in 2004. His next assignment was to the Air Force Operational Test and Evaluation Center at Kirtland Air Force Base, New Mexico. In 2008, he deployed in support of the Global War on Terrorism to the Combined Air Operations Center at Al-Udeid Air Base, Qatar. In 2009, he was selected to pursue a doctorate in Operations Research from the Air Force Institute of Technology at WPAFB, Ohio. Following graduation in 2012, he will be stationed at Fort Lee, Virginia where he will be teaching the Operations Research/Systems Analysis Military Applications Course at the Army Logistics University.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)* 30–09–2012	**2. REPORT TYPE** Doctoral Dissertation	**3. DATES COVERED** *(From — To)* Aug 2009 — Sep 2012

4. TITLE AND SUBTITLE Minimization of Collateral Damage in Airdrops and Airstrikes	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Dillenburger, Steven P., Major, USAF	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/DS/ENS/12S-01
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Intentionally left blank	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Collateral damage presents a significant risk during air drops and airstrikes, risking citizens' lives and property, straining the relationship between the United States Air Force and host nations. This dissertation presents a methodology to determine the optimal location for making supply airdrops in order to minimize collateral damage while maintaining a high likelihood of successful recovery. A series of non-linear optimization algorithms is presented along with their relative success in finding the optimal location in the airdrop problem. Additionally, we present a quick algorithm for accurately creating the Pareto frontier in the multi-objective airstrike problem. We demonstrate the effect of differing guidelines, damage functions, and weapon employment selection which significantly alter the location of the optimal aimpoint in this targeting problem. Finally, we have provided a framework for making policy decisions in fast-moving troops-in-contact situations where observers are unsure of the nature of possible enemy forces in both finite horizon and infinite horizon problems. Through the recursive technique of solving this Markov decision process we have demonstrated the effect of improved intelligence and differing weights for waiting and incorrect decisions in the face of uncertain situations.

15. SUBJECT TERMS

Bivariate probability, non-linear optimization, multi-objective optimization, evolutionary algorithms, Markov decision process

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Jeffery K. Cochran
U	U	U	UU	130	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255–3636, x4521; jeffery.cochran@afit.edu